

**ANALISIS PERMASALAHAN PADA APLIKASI
SMARTPHONE BPJSTKU MENGGUNAKAN METODE
FISHBONE DIAGRAM DAN TEXT MINING**

TUGAS AKHIR

**Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
Sarjana Strata-1 Pada Jurusan Teknik Industri Fakultas
Teknologi Industri**



Nama : Huzein Mulyawan

Nim : 16522111

**PROGRAM STUDI TEKNIK INDUSTRI
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA**

2021

PERNYATAAN KEASLIAN TUGAS AKHIR

PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertandatangan di bawah ini:

Nama : Huzein Mulyawan

Nim : 16522111

Dengan tugas akhir berjudul :

ANALISIS PERMASALAHAN PADA APLIKASI *SMARTPHONE* BPJSTKU MENGGUNAKAN METODE *FISHBONE* DIAGRAM DAN *TEXT MINING*.

Saya sebagai penulis menyatakan bahwa tugas akhir ini merupakan hasil penelitian saya sendiri dengan beberapa kutipan yang telah saya cantumkan sumbernya. Bila di kemudian hari ditemukan kesamaan ataupun penyimpangan yang merugikan orang lain, saya bersedia menerima konsekuensi yang diberikan oleh Universitas Islam Indonesia dengan penuh rasa tanggung jawab.

Yogyakarta, 20 Oktober 2021

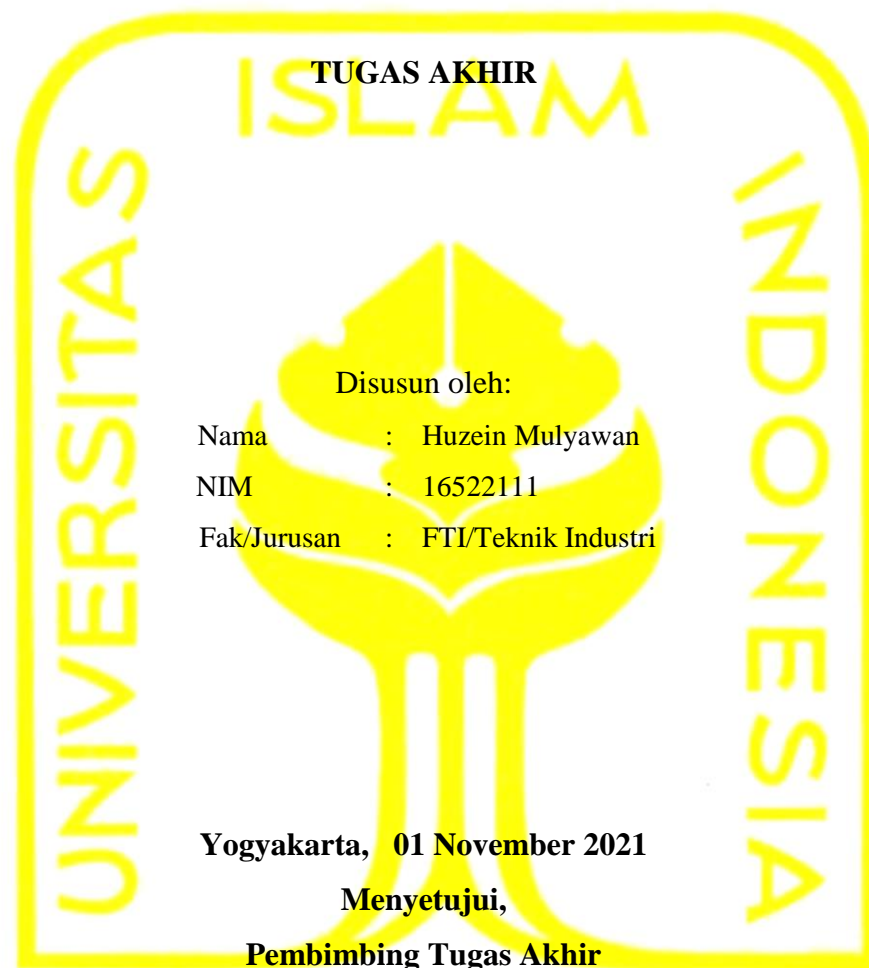


(Huzein Mulyawan)

16522111

LEMBAR PENGESAHAN PEMBIMBING

**ANALISIS PERMASALAHAN PADA APLIKASI *SMARTPHONE* BPJSTKU
MENGUNAKAN METODE *FISHBONE DIAGRAM* DAN *TEXT MINING***



الإسلامية الإسلامية
الجامعة الإسلامية
الاندونيسية

(Andrie Pasca Hendradewa, ST., M.T.)

NIP. 155220110

LEMBAR PENGESAHAN PENGUJI

LEMBAR PENGESAHAN PENGUJI

ANALISIS PERMASALAHAN PADA APLIKASI *SMARTPHONE* BPJSTKU
MENGUNAKAN METODE *FISHBONE DIAGRAM* DAN *TEXT MINING*

Disusun oleh:

Nama : Huzain Mulyawan

NIM : 16522111

Fak/Jurusan : FTI/Teknik Industri

Telah dipertahankan di depan sidang penguji sebagai satu syarat untuk
memperoleh gelar Sarjana Strata-1 Teknik Industri

Yogyakarta, 2021

Tim Penguji

Andrie Pasca Hendradewa, ST., M.T.

Ketua

M. Ridwan Andi Purnomo, S.T., M.Sc., Ph.D

Anggota I

Ir. Ira Promasanti Rachmadewi, M.Eng.

Anggota II

Mengetahui

Ketua Program Studi Teknik Industri

Universitas Islam Indonesia



Huzain Mulyawan, S.T., M.M.

HALAMAN PERSEMBAHAN

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Tugas akhir ini saya persembahkan untuk kedua orang tua, kakak saya dan seluruh pihak yang telah membantu dan memotivasi saya untuk menyelesaikan tugas akhir ini.

Tidak terkecuali untuk saya persembahkan kepada seluruh dosen dan karyawan Universitas Islam Indonesia yang telah memberi ilmu dan membimbing ke jalan yang baik selama saya kuliah di Universitas Islam Indonesia

الجمعة المباركة
الاستاذة الدكتورة

HALAMAN MOTTO

مَنْ سَلَكَ طَرِيقًا يَتَمَسُّ فِيهِ عِلْمًا، سَهَّلَ اللَّهُ لَهُ طَرِيقًا إِلَى الْجَنَّةِ . رَوَاهُ مُسْلِمٌ

"Barang siapa menempuh satu jalan (cara) untuk mendapatkan ilmu, maka Allah pasti mudahkan baginya jalan menuju surga." (HR. Muslim)

إِذَا مَاتَ الْإِنْسَانُ انْقَطَعَ عَنْهُ عَمَلُهُ إِلَّا مِنْ ثَلَاثَةٍ إِلَّا مِنْ صَدَقَةٍ جَارِيَةٍ أَوْ عِلْمٍ يُنْتَفَعُ بِهِ أَوْ وَلَدٍ صَالِحٍ يَدْعُو لَهُ

"Jika seorang manusia mati, maka terputuslah darinya semua amalnya kecuali dari tiga hal; dari sedekah jariyah atau ilmu yang diambil manfaatnya atau anak shalih yang mendoakannya." (HR. Muslim no. 1631)

KATA PENGANTAR

Assalamualaikum warrahmatullah wabarakatuh

Alhamdulillah segala puji dan syukur saya panjatkan kepada Allah subhannahuwataa'la atas nikmat dan karunia yang diberikan berupa sehat fisik dan rohani untuk bisa menyelesaikan laporan tugas akhir **“ANALISIS PERMASALAHAN PADA APLIKASI SMARTPHONE BPJSTKU MENGGUNAKAN METODE FISHBONE DIAGRAM DAN TEXT MINING”** tepat pada waktunya. Laporan tugas akhir ini disusun untuk memenuhi tugas mata kuliah tugas akhir bagi mahasiswa di Prodi Teknologi Industri Jurusan Teknik Industri Universitas Islam Indonesia Yogyakarta.

Tugas akhir ini dilakukan sebagai salah satu persyaratan yang harus dipenuhi dalam menyelesaikan jenjang Strata-1 di Jurusan Teknik Industri Universitas Islam Indonesia.. Pada kesempatan ini penulis ingin menyampaikan ucapan terima kasih kepada seluruh pihak yang telah membantu dalam menyelesaikan laporan ini, yaitu kepada :

1. Bapak Prof. Dr. Ir. Hari Purnomo, M.T. selaku Dekan Fakultas Teknologi Industri Universitas Islam Indonesia.
2. Bapak Dr. Taufiq Immawan, S.T., M.M. selaku Ketua Program Studi Teknik Industri Fakultas Teknologi Industri Universitas Islam Indonesia.
3. Bapak M. Ridwan Andi Purnomo, S.T., M.Sc., Ph.D. selaku Ketua Jurusan Teknik Industri Universitas islam indonesia.
4. Bapak Andrie Pasca Hendradewa, ST., M.T selaku Dosen Pembimbing Tugas Akhir yang membantu membimbing penulis menyelesaikan tugas akhir skripsi.
5. Bapak, Ibu dan Kakak penulis yang selalu mendoakan dan mendukung selalu agar tugas akhir dapat selesai.
6. Teman dekat dan teman angkatan 2016 teknik industri universitas islam indonesia yang mendoakan, dan mendukung agar tugas akhir dapat segera selesai.
7. Ibu kos penulis yang juga turut serta memberi semangat bagi penulis untuk dapat menyelesaikan Tugas akhir.

Dalam penulisan Tugas Akhir ini, penulis menyadari bahwa masih terdapat kekurangan yang perlu diberikan kritik dan saran yang membangun. Oleh karena itu, penulis memohon maaf yang sebesar-besarnya kepada pihak yang terlibat serta pembaca Tugas Akhir ini. Namun terlepas dari itu semua, penulis berharap Tugas Akhir ini dapat memberikan manfaat bagi penulis dan para pembaca.

Wassalamu'alaikum warahmatullahi wabarakatuh

Yogyakarta, 20 Oktober 2021



(Huzein Mulyawan)



ABSTRAK

Pada beberapa tahun terakhir Indonesia mulai merubah secara bertahap revolusi industri 3.0 menjadi 4.0 dengan tujuan bisa berkompetisi dengan negara lain di bidang industri. Agar perubahan tersebut bisa terwujud, pemerintah kemudian melakukan perubahan di pemerintahan yang bernama pemerintahan elektronik atau *E-government*. Salah satu peran pemerintah dalam mewujudkan *e-government* yaitu pembuatan aplikasi *smartphone* bernama BPJSTKU di badan usaha milik negara bagian asuransi yang biasa disebut badan penyelenggara jaminan sosial ketenagakerjaan atau BPJS Ketenagakerjaan. Aplikasi BPJSTKU disambut baik oleh masyarakat, karena mempermudah dalam mengakses informasi seperti saldo tabungan serta tidak perlu memakan waktu lama dalam mengambil informasi tersebut. Akan tetapi, aplikasi BPJSTKU tidak lepas dari ulasan pengguna aplikasi yang menyuarakan terdapat beberapa masalah didalam BPJSTKU. Untuk membantu pemerintah dalam mengembangkan teknologi informasi, maka dilakukan penelitian untuk mengetahui penyebab dari permasalahan yang ada dalam BPJSTKU serta memberi solusinya. Penelitian menggunakan metode *text mining* dan *fishbone diagram*. Hasil dari *text mining* yang didapat dari Google Playstore berjumlah 48931 dengan peringkat 1 dan 2 yaitu berupa lima kata kunci yang menyebabkan beberapa masalah di aplikasi BPJSTKU seperti “aplikasi”, “login”, “daftar”, “email”, “bpjstku” serta hasil hubungan korelasi antara kata kunci dengan kata lain yang menunjukkan suatu pola. Kemudian hasil tersebut dimasukkan kedalam diagram *fishbone* untuk diketahui secara detail penyebab permasalahan. Beberapa penyebab permasalahan yang ditemukan yakni *update* yang bermasalah dan kode verifikasi yang bermasalah. Solusi yang dilakukan yaitu diperlukan ahli yang sesuai dalam bidangnya untuk melakukan perbaikan pada aplikasi BPJSTKU secara *trial & error* dibagian data *server* dan kode pemrograman.

Kata kunci: Industri 4.0, *E-government*, *Text mining*, *Fishbone diagram*, BPJSTKU

DAFTAR ISI

TUGAS AKHIR	i
PERNYATAAN KEASLIAN TUGAS AKHIR	1
LEMBAR PENGESAHAN PEMBIMBING.....	2
LEMBAR PENGESAHAN PENGUJI.....	3
HALAMAN PERSEMBAHAN	4
HALAMAN MOTTO.....	5
KATA PENGANTAR	6
ABSTRAK.....	8
DAFTAR ISI.....	9
DAFTAR TABEL.....	12
DAFTAR GAMBAR.....	13
BAB 1 PENDAHULUAN	14
1.1 Latar Belakang	14
1.2 Rumusan Masalah	16
1.3 Batas Permasalahan.....	16
1.4 Tujuan Penelitian	17
1.5 Manfaat Penelitian	17
1.6 Sistematika Penulisan.....	17
BAB 2 KAJIAN LITERATUR.....	19
2.1 Kajian Deduktif.....	19
2.1.1 E-Government.....	19
2.1.2 BPJS Ketenagakerjaan	19
2.1.3 <i>Key Performance Indicator of Mobile Application</i>	21
2.1.4 <i>Machine Learning</i>	22

2.1.5	<i>Text Mining</i>	22
2.1.6	<i>Word cloud</i>	27
2.1.7	<i>Association Rules</i>	28
2.1.8	<i>Fishbone diagram</i>	30
2.2	Kajian Induktif	31
BAB 3	METODE PENELITIAN	38
3.1	Jenis Data	38
3.2	Prosedur Penelitian.....	38
3.2.1	Tahap Persiapan Data	38
3.2.2	Tahap Pengumpulan Data.....	39
3.2.3	Tahap Pengolahan Data	39
3.3	<i>Diagram Alur</i>	42
BAB 4	PENGOLAHAN DATA.....	46
4.1	Text Mining.....	46
4.1.1	<i>Text Preprocessing</i>	46
4.1.2	<i>Data Mining</i>	49
4.2	WordCloud.....	51
4.3	<i>Fishbone Diagram</i>	53
BAB 5	PEMBAHASAN.....	59
5.1	Text Mining.....	59
5.2	Wordcloud.....	60
5.3	<i>Fishbone diagram</i>	61
BAB 6	KESIMPULAN	63
6.1	Kesimpulan	63
6.2	Saran.....	64
	DAFTAR PUSTAKA	65
	LAMPIRAN.....	70



DAFTAR TABEL

Tabel 1 Penelitian Terdahulu.....	35
Tabel 2 Hasil <i>association rules</i> untuk 5 kata.....	54



DAFTAR GAMBAR

Gambar 1 BPJSTKU di Playstore.....	15
Gambar 2 Ulasan dan Rating	16
Gambar 3 <i>Part of Speech Tagging</i>	24
Gambar 4 <i>Parsing Tree Chunks</i>	24
Gambar 5 <i>Bag of words</i>	25
Gambar 6 <i>Bag of words dengan Vector space</i>	26
Gambar 7 <i>Word cloud</i>	27
Gambar 8 <i>Fishbone diagram</i>	30
Gambar 9 Diagram Alur	42
Gambar 10 Diagram Alur Hasil dan Pengolahan Data.....	44
Gambar 11 Data mentah ke <i>Text Normalization</i> Kata.....	47
Gambar 12 Data Mentah ke <i>Text Normalisasi</i> Frasa.....	48
Gambar 13 <i>Text normalization</i> ke <i>Tokenization</i>	48
Gambar 14 <i>Tokenization</i> ke <i>Multidimensional list</i>	49
Gambar 15 <i>Multidimensional list</i> ke Pelabelan <i>true & false</i>	50
Gambar 16 Pelabelan <i>true & false</i> ke <i>Association Rules</i>	50
Gambar 17 <i>Tokenization</i> ke <i>wordcloud</i>	52
Gambar 18 <i>Bar chart</i>	53
Gambar 19 <i>Fishbone diagram</i>	58
Gambar L. 1 Halaman Utama AppFollow	70
Gambar L. 2 Memilih Aplikasi.....	70
Gambar L. 3 Menuju Reviews Para Pengguna BPJSTKU	71
Gambar L. 4 Memilih Tanggal Untuk Data Reviews yang kemudian di Export	71
Gambar L. 5 Download All Reviews dan Pilih CSV	72

BAB 1

PENDAHULUAN

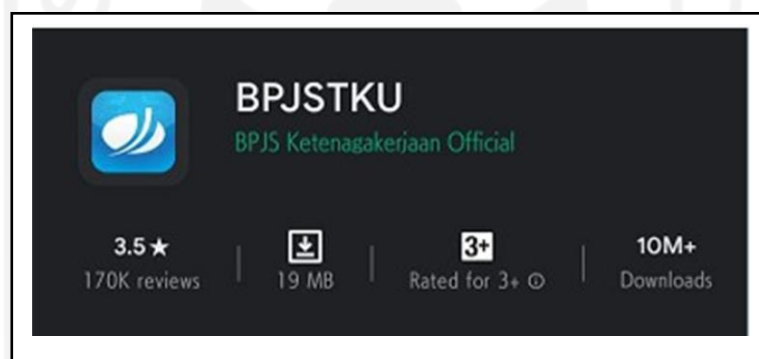
1.1 Latar Belakang

Beberapa tahun terakhir ini perkembangan teknologi informasi dan komunikasi di Indonesia begitu cepat khususnya perkembangan jumlah pengguna internet dan komunikasi telepon seluler. Tercatat pada tahun 2019 bahwa 47,69 % populasi masyarakat Indonesia mulai menggunakan internet (BPS, 2020). Pada tahun 2020 pengguna internet mencapai 58,2 % dan kemudian pada tahun 2021 pengguna internet menjadi 73,7 % atau sekitar 202,6 juta jiwa (Riyanto, 2021). Untuk perkembangan telepon seluler khususnya *smartphone* tercatat bahwa pada tahun 2020, 49 % populasi masyarakat Indonesia menggunakan *smartphone* (Rakhmayanti, 2020). Dalam *smartphone* yang digunakan masyarakat Indonesia, *operating system/OS* yang dipakai sebagian besar adalah Android yaitu mencapai 91 % (Nurhayati-Wolff, 2021).

Badan penyelenggaraan jaminan nasional ketenagakerjaan atau disingkat BPJS Ketenagakerjaan merupakan sebuah lembaga hukum yang bertugas untuk melindungi seluruh pekerja yang ada di Indonesia (Tempo, 2021). BPJS Ketenagakerjaan memiliki 4 program yaitu berupa Jaminan Kecelakaan Kerja yang disingkat JKK, Jaminan Kematian disingkat JK, Jaminan Pensiun atau JP dan yang terakhir Jaminan Hari Tua atau bisa dipanggil JHT. Program pertama yaitu JKK memiliki arti bahwa para pekerja akan mendapat perlindungan, jika terdapat kecelakaan dalam bekerja. Program kedua yaitu JK berarti pekerja yang meninggal dunia mendapatkan uang tunai dan beasiswa untuk keluarga yang ditinggalkan. Program ketiga yaitu JP mengandung arti pekerja mendapatkan jaminan pensiun, agar pekerja bisa hidup layak setelah pensiun. Program keempat yaitu JHT yang memiliki fungsi yang hampir sama seperti JP, perbedaannya

terletak pada cara iuran uang dan jaminan uang yang didapatkan (Nuraini, 2021). Dalam pelaksanaan tugasnya, BPJS Ketenagakerjaan mendapatkan beberapa masalah berupa peningkatan cakupan peserta, peningkatan layanan dan manfaat bagi pekerja, dan optimilasi dana investasi. Dari permasalahan tersebut dan ditambah dengan perkembangan teknologi informasi dan komunikasi di Indonesia begitu cepat, BPJS Ketenagakerjaan mendapatkan peluang untuk menangani masalah tersebut dengan cara merubah pelayanan manual menjadi digital (Lumanauw, 2021).

Perubahan layanan manual dan digital sudah mulai dilaksanakan oleh BPJS Ketenagakerjaan di antara lain pembuatan aplikasi bernama BPJSTKU. BPJSTKU berfungsi sebagai pemberi informasi tentang rincian saldo JHT, lihat kartu dan saldo JHT, kirim saldo JHT, dan simulasi perhitungan JHT (Nurrahmi, 2018). Informasi aplikasi BPJSTKU yang didapat dari Google Playstore bisa dilihat di gambar berikut :



Gambar 1 BPJSTKU di Playstore

Pada gambar 1 terdapat peringkat 3, 5 dan jumlah ulasan sekitar 170,000 ribu. Dalam ulasan tersebut terdapat pengguna yang hanya memberikan peringkat tanpa ulasan / feedback. Untuk mendapatkan pengguna yang melakukan ulasan sekaligus peringkat digunakanlah aplikasi bernama AppFollow yang mana data yang digunakan dari sejak aplikasi BPJSTKU masuk di Google Playstore sampai tanggal 11 Juli 2021 saat pengambilan data dimulai. Hasil yang didapat berjumlah total 86411 ulasan dan peringkat keseluruhan. Dari 86411 ulasan dengan peringkat sebesar 2,59 yang mana peringkat dari angka 1 yang berarti sangat buruk sampai 5 yang berarti sangat baik. Berikut gambar 2 yang menunjukkan hasil dari AppFollow :



Gambar 2 Ulasan dan Rating

sumber: AppFollow

Terlihat pada gambar 2 bahwa masih banyak pengguna yang memberi ulasan dan peringkat baru yang belum memuaskan, Sedangkan aplikasi BPJSTKU merupakan salah satu cara pekerja bisa mendapatkan informasi lengkap tentang 4 program dengan mudah melalui aplikasi di *smartphone* dibandingkan lewat sms yang kurang lengkap atau secara *offline* dengan mengunjungi kantor BPJS Ketenagakerjaan (Idris, 2021).

1.2 Rumusan Masalah

1. Bagaimana cara mengetahui permasalahan yang dialami pengguna di aplikasi BPJSTKU?
2. Bagaimana solusi perbaikan dari masalah pada aplikasi BPJSTKU?

1.3 Batas Permasalahan

Batas permasalahan dalam penelitian ini adalah :

1. Ulasan dan Peringkat diambil di Google Playstore.
2. Pengambilan data dilakukan pada tanggal 28 februari 2018 sampai 11 juli 2021.
3. Penelitian dilakukan untuk menemukan masalah yang ada di aplikasi BPJSTKU.
4. Tidak merambah ke sistem informasi dan manajerial.

1.4 Tujuan Penelitian

1. Tujuan dari penelitian ini untuk mengetahui masalah yang dialami pengguna di aplikasi BPJSTKU.
2. Memberikan usulan perbaikan pada aplikasi BPJSTKU.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini yaitu memberi tahu beberapa permasalahan yang dialami pengguna aplikasi BPJSTKU dan rekomendasi perbaikan agar penelitian selanjutnya bisa melanjutkan pengembangan aplikasi ketahap berikutnya tanpa mencari permasalahan pengguna aplikasi BPJSTKU terlebih dahulu.

1.6 Sistematika Penulisan

Sistematika penulisan dibuat untuk mengetahui tahapan – tahapan dalam penelitian serta mengetahui maksud dari tahapan itu, berikut tahapan penelitian ini :

- **BAB 1 PENDAHULUAN**
Menjelaskan Latar belakang, Rumusan masalah, Batas Permasalahan Tujuan Penelitian, Manfaat Penelitian dan Sistematika Penulisan dalam penelitian ini.
- **BAB 2 KAJIAN LITERATUR**
Menjelaskan isi yang ada di dalam kajian deduktif dan induktif dalam penelitian ini.
- **BAB 3 METODE PENELITIAN**
Memaparkan dan menjelaskan Jenis data, prosedur penelitian dan *Diagram* alir yang ada di penelitian ini.
- **BAB 4 PENGUMPULAN DAN PENGOLAHAN DATA**
Memaparkan hasil dan Menjelaskan hasil pada pengumpulan dan pengolahan data yang ada di penelitian ini.
- **BAB 5 PEMBAHASAN**

Menjelaskan hasil dari bab 4 lebih mendalam.

- **BAB 6 KESIMPULAN**

Memaparkan kesimpulan dari hasil bab sebelumnya yang kemudian

Menjelaskan inti permasalahan dari penelitian ini.



BAB 2

KAJIAN LITERATUR

Kajian literatur terdapat dua bagian yaitu kajian deduktif dan kajian induktif. Kajian deduktif diisi penjelasan tentang *text mining*, *wordcloud*, *fishbone diagram*, dan *association rule*. Untuk kajian induktif diisi dengan laporan penelitian sebelumnya menggunakan metode yang sebagian sama dengan penelitian ini, tetapi kasusnya berbeda.

2.1 Kajian Deduktif

2.1.1 E-Government

Mirchandani et al. (2008) mengatakan bahwa *e-government* merupakan suatu usaha pemerintah untuk memperkuat hubungan antara pemerintah dengan masyarakat dan bisnis menggunakan cara meningkatkan kualitas teknologi informasi komunikasi menjadi lebih murah dan cepat untuk layanan jasa, informasi dan pengetahuan. Selain itu *e-government* bisa meningkatkan tata pengelolaan pemerintah menjadi lebih baik karena adanya transparansi dari efek *e-government*. Selain untuk meningkatkan hubungan, *e-government* dianggap oleh pemerintahan indonesia sebagai rencana yang paling cocok digunakan di wilayah kepulauan dan populasi yang besar (Sabani, 2018). Diketahui pada tahun 2019 pengguna *mobile internet* mencapai 61,85% dari jumlah kelseluruhan populasi (Nurhayati-Wolff, *Mobile Internet & Apps*, 2020). Untuk populasi masyarakat indonesia mencapai angka 270 juta jiwa (BPS, Hasil Sensus Penduduk 2020, 2021).

2.1.2 BPJS Ketenagakerjaan

Badan Penyelenggara Jaminan Sosial Ketenagakerjaan atau BPJSTK (BPJS Ketenagakerjaan) adalah sebuah badan hukum untuk publik dengan tujuan memberikan

perlindungan kepada seluruh tenaga kerja Indonesia dalam bentuk asuransi sosial (BUMN, 2021). BPJS Ketenagakerjaan mempunyai beberapa program untuk dapat merealisasikan tujuan UU NO 24 Tahun 2011. Berikut beberapa Program BPJS Ketenagakerjaan untuk mencapai tujuannya (Cermati, 2019), yakni :

1. Jaminan Kecelakaan Kerja (JKK)

BPJS Ketenagakerjaan akan memberi perlindungan pekerja dari resiko kecelakaan kerja dan penyakit yang disebabkan oleh lingkungan kerja.

2. Jaminan Kematian (JKM)

BPJS Ketenagakerjaan akan memberikan manfaat berupa uang tunai kepada ahli waris ketika pekerja meninggal dunia akibat lingkungan kerja.

3. Jaminan Hari Tua (JHT)

BPJS Ketenagakerjaan akan memberikan manfaat berupa uang tunai dari hasil akumulasi iuran dan dana pengembangan.

4. Jaminan Pensiun (JP)

BPJS Ketenagakerjaan akan memberikan penghasilan kepada para pekerja yang sudah pensiun atau ahli waris.

Keempat program tersebut selain untuk merealisasikan tujuan UU No 24 tahun 2011 dapat juga untuk melaksanakan misi dan visi BPJS Ketenagakerjaan. (Putri, 2020) Mengatakan misi BPJS Ketenagakerjaan terhitung berjumlah tiga yakni :

1. Melindungi dan mensejahterakan seluruh pekerja dan keluarganya.
2. Meningkatkan produktivitas dan daya saing.
3. Mendukung pembangunan dan kemandirian perekonomian sosial.

Untuk visi, BPJS Ketenagakerjaan sendiri menginginkan bahwa BPJS Ketenagakerjaan mampu menjadi badan penyelenggara jaminan sosial kebanggaan bangsa, yang amanah, bertata kelola baik serta unggul dalam operasional dan pelayanan. Menurut Nisa (2020) pada tanggal 23 Mei 2017 BPJS Ketenagakerjaan mengeluarkan aplikasi bernama BPJSTK *Mobile* yang kemudian berubah nama menjadi BPJSTKU. Menurut Dzulfikar (2019) BPJSTKU dibuat untuk mempermudah pekerja dalam memanfaatkan BPJS Ketenagakerjaan dengan fitur – fitur sebagai berikut :

1. Pendaftaran BPJS Ketenagakerjaan secara daring.
2. Cek saldo JHT BPJS Ketenagakerjaan.
3. Simulasi perhitungan JHT.
4. Pelaporan jika peserta terjadi kecelakaan secara *realtime*.

5. Pengaduan tentang pelayanan BPJS Ketenagakerjaan.
6. Melihat dan mengunduh kartu peserta digital BPJS Ketenagakerjaan.
7. Informasi program dan layanan BPJS Ketenagakerjaan.
8. Informasi cabang terdekat.
9. Informasi pusat layanan dan media sosial.

2.1.3 *Key Performance Indicator of Mobile Application*

Menurut KPI (2022) *key performance indicator* adalah indikator penting sebuah proses yang mana proses tersebut memiliki hasil yang diinginkan. Sedangkan *mobile application* berarti kumpulan beberapa program yang dapat di eksekusi atau di jalankan disebuah peralatan selular dan melakukan tugas tertentu untuk pengguna (Islam, 2010). Jadi *key performance indicator of mobile application* adalah indikator penting sebuah proses yang mana proses tersebut memiliki hasil yang diinginkan di sebuah aplikasi telepon selular. Menurut Smith (2019) *key performance indicator of mobile application* memiliki alat ukur kinerja inti untuk aplikasi telepon seluler. Alat ukur tersebut di antaranya yaitu:

1. *App Load Period*

Alat ukur kinerja tentang seberapa mampu aplikasi dapat mengatasi banyak permintaan tanpa menurunkan kinerja aplikasi.

2. *API Latency*

Alat ukur kinerja tentang seberapa cepat aplikasi melakukan perjalanan pulang pergi yang dimulai dari mengirimkan permintaan hingga menghasilkan respon.

3. *App Crash*

Alat ukur kinerja tentang seberapa sering aplikasi mengalami kegagalan saat sedang *loading*.

4. *Network Errors*

Alat ukur kinerja tentang seberapa sering aplikasi mengalami "HTTP error" , hal ini dapat menyebabkan aplikasi kegagalan atau waktu respon yang lama.

2.1.4 *Machine Learning*

Machine learning adalah bidang yang mengembangkan algoritma atau perhitungan matematika yang didesain bisa diaplikasikan dalam bentuk dataset, dengan area fokus disekitar *regression*, *classification*, dan *clustering* (Athey, 2019). Menurut IBM (2020), *machine learning* adalah sebuah cabang dari *artificial intelligence* (AI) yang berfokus membuat aplikasi lalu belajar dari data yang diberikan dan kemudian mengembangkan tingkat akurasi secara berkala tanpa diprogram melakukannya, dengan beberapa metode diantaranya yaitu:

1. *Supervised learning*

Metode yang menggunakan data berlabel untuk melatih diri sendiri. Data berlabel ini di pisah menjadi dua bagian yaitu data *train* dan data *test*. data *train* dibuat sebagai dasar untuk memprediksi data *test* atau data yang akan kita prediksi.

2. *Unsupervised learning*

Metode yang menggunakan data tidak berlabel yang mana algoritma bekerja untuk mengambil fitur yang dibutuhkan lalu fitur akan diberi label dan mengelompokkan secara *real time*.

3. *Semi supervised learning*

Metode yang menggunakan sedikit data yang berlabel untuk mengarahkan pengelompokkan dan pengambilan fitur dari data berskala besar yang tidak berlabel.

2.1.5 *Text Mining*

Text mining adalah salah satu cabang variasi dari bidang data *mining* untuk menemukan pola menarik di dalam *database* yang berukuran besar. Pengertian *text mining* hampir sama dengan data *mining*, tetapi ada letak perbedaan di antara kedua tersebut yang bisa dilihat dari jenis data yang digunakan. Untuk *text mining*, pengambilan pola informasi didapat dari *database* tekstual dan data bersifat *Natural Language Processing* atau dengan kata lain data yang akan diambil dalam bentuk teks yang dapat dibaca oleh manusia, namun untuk data *mining* pengambilan pola menarik dilakukan di data yang terstruktur (Hearst, 2003).

Menurut Mary (2020), secara umum proses *text mining* memiliki 5 tahap proses, berikut prosesnya :

1. *Text pre-Processing* : data awal yang didapat masih mentah, maka perlu dibersihkan. Terdapat beberapa tahapan dalam melakukan *pre-processing*, berikut tahapannya :

- *Text Normalization*

Pada tahap ini melakukan perubahan data menjadi format standart. Prosesnya seperti pengubahan bentuk teks menjadi huruf kapital semua atau kecil semua, kemudian tanda baca, emoji, dan kata yang dikira tidak berguna dibuang.

- *Tokenization*

Pada proses ini kalimat yang terdapat di teks akan dipecah menjadi per kata contoh : “dia liburan ke wilayah utara” menjadi “dia”, “liburan”, “ke”, “arah”, “utara”. Menurut Chakravarthy (2020), Tujuan perubahan kalimat ke perkata yaitu untuk membantu penganalisaan teks menjadi lebih mudah.

- *Stemming*

Proses *stemming* yaitu proses untuk mengembalikan suatu kata ke bentuk paling dasar. Menurut Heidenreich (2018) *stemming* adalah proses penyederanaan sebuah kata untuk mendekati kata asal tersebut. contoh : “memesan” menjadi “mesan” kata imbuan prefik –me dihilangkan.

- *Lemmatization*

Proses yang tujuannya mirip seperti *stemming*, tetapi pensederhanaan kata dibuat semirip mungkin seperti kata asal tersebut. untuk melakukannya diperlukan data yang banyak untuk mendapatkan kata asal agar kata yang akan diubah bisa kembali ke kata asal dengan benar.

- *Part of Speech Tagging*

Proses yang memberikan tanda unsur pada setiap kata berdasarkan arti dan konteks kata itu sendiri. Berikut contoh penggunaan *part of speech tagging* dalam bentuk gambar menurut Pykes (2020):

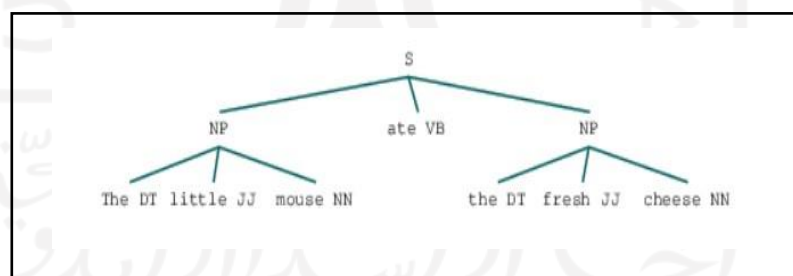


Gambar 3 *Part of Speech Tagging*

Sumber : (Pykes, 2020)

- *Chunks*

Berbeda dengan proses *part of speech tagging* yang menandai unsur pada tingkat kata, *chunking* dilakukan pada tingkat frasa. Tujuan dari penandaan ini untuk mendapatkan makna gramatikal dari sebuah kalimat. Menurut Bachani (2020), secara singkat *chunking* berarti mengelompokkan beberapa kata/token menjadi sebuah potongan kalimat. Berikut contoh *chunks* dalam bentuk ilustrasi gambar:



Gambar 4 *Parsing Tree Chunks*

Sumber : (Bachani, 2020)

- *Named Entity Recognition*

Pada proses ini bertujuan untuk menemukan entitas nama di dalam teks dan mengklasifikasikannya ke bentuk kategori yang telah ditentukan sebelumnya.

- *Relationship Extraction*

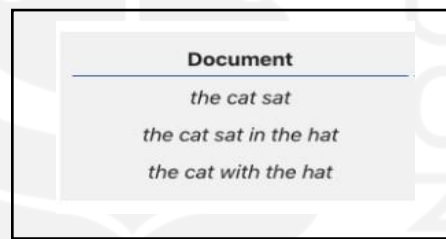
Proses ini perkembangan dari proses *Named Entity Recognition*, dengan mengidentifikasi relasi antara *named entities* seperti nama orang, organisasi, dll.

2. *Text Transformation*

Proses ini fokus pada representasi dari isi teks yang akan digunakan serta jumlah isi teks yang terjadi. Terdapat dua pendekatan untuk menggunakan proses *text transformation* yaitu berupa:

- *Bag of Words*

Proses representasi teks menjadi multi set, mengabaikan tata bahasa dan bahkan urutan bahasa. Berikut contoh gambar ilustrasinya :



Gambar 5 *Bag of words*

Sumber: (Zhou, 2019)

Pada gambar 5 terlihat tiga buah kalimat dengan struktur kata yang berbeda. Ketiga kalimat tersebut nantinya akan diubah menjadi sebuah angka atau biasa disebut *vectorization* di proses pendekatan selanjutnya.

- *Vector Space*

Proses ini merupakan kelanjutan dari proses *bag of words* yang mana teks dirubah dalam bentuk angka yang biasa dinamakan vektor. Angka bisa bertambah sesuai dengan tingkat kemunculan setiap kata dari sebuah kalimat di dalam teks. Dari angka bisa diketahui nantinya yang mana kata yang penting disebuah teks dengan dilihat jumlah total perkata dalam teks. Berikut contoh gambar ilustrasi :

Document	the	cat	sat	in	hat	with
<i>the cat sat</i>	1	1	1	0	0	0
<i>the cat sat in the hat</i>	2	1	1	1	1	0
<i>the cat with the hat</i>	2	1	0	0	1	1

Gambar 6 *Bag of words dengan Vector space*

Sumber : (Zhou V. , 2019)

Pada gambar 6 terlihat kata “the” mempunyai jumlah angka paling banyak dari semua kata yang diambil dalam tiga buah kalimat berbeda secara otomatis kata “the” adalah kata yang paling penting dalam 3 kalimat yang berbeda tersebut.

3. *Feature Selection*

Pada tahap ini dilakukan pemilihan fitur atau atribut atau variabel yang paling relevan dari variabel yang tersedia sehingga memberikan informasi secara maksimal kepada variabel prediksi. Variabel yang tidak relevan akan meningkatkan kompleksitas dan mengurangi akurasi dalam analisa.

4. *Data Mining*

Pada tahap ini setelah membuat data dari yang tidak terstruktur menjadi data terstruktur kemudian dilakukan proses data *mining* untuk mengambil informasi dari data terstruktur. Teknik yang digunakan dalam proses data *mining* berupa *classification, clustering, regression, outlier, sequential patterns, association rules* dan prediksi.

5. *Evaluation*

Setelah melakukan data *mining* dilanjutkan ke bagian evaluasi. Proses evaluasi digunakan untuk menilai hasil akhir data *mining* yang didapat serta melakukan cek akurasi pada hasil tersebut.

2.1.6 Word cloud

Menurut Jin (2017), *word cloud* merupakan suatu jenis daftar yang memiliki bobot yang kemudian digambarkan kedalam bahasa atau teks data, untuk mengetahui kata yang memiliki bobot lebih besar dari kata lainnya dengan memperbesar ukuran *font* kata. diketahui terdapat tiga metode *word cloud* dalam segi algoritma yakni:

1. *Frequency*

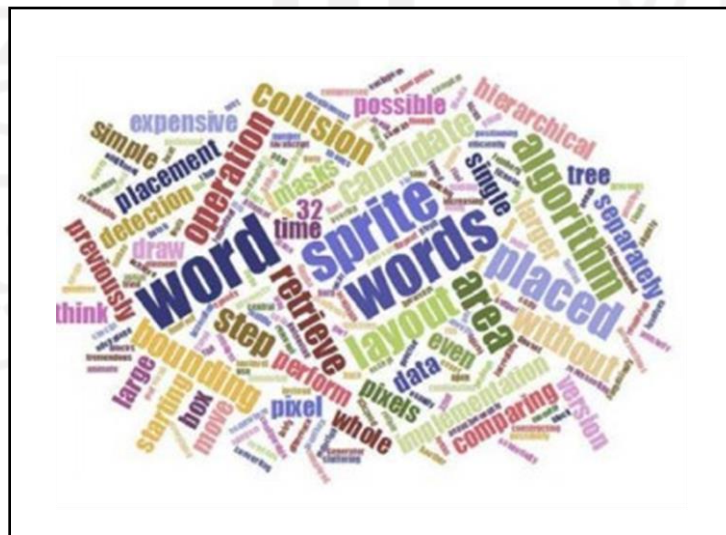
Metode yang menilai bobot atau mengubah ukuran besar kecil *font* kata didasarkan dari jumlah kata yang muncul dari data teks. Metode ini biasa digunakan di *text mining*.

2. *Categorization*

Dalam metode *categorization*, perubahan ukuran *font* kata menunjukkan subkategori dari data peta geografi. Metode ini digunakan di pemetaan geografi.

3. *Mixed*

Dalam metode *mixed*, menggunakan gabungan metode *frequency* dan *categorization*, yang mana membutuhkan analisis logikal di data yang kompleks sebelum menyiapkan peta *word cloud*.



Gambar 7 Word cloud

Sumber : Jin (2017)

2.1.7 Association Rules

Association rules adalah teknik untuk menemukan hubungan menarik antar *item* diantara sekumpulan data besar (Najafabadi, 2017). Menurut Hahsler (2017) *association rules* adalah salah satu teknik utama untuk mendeteksi dan mengambil informasi yang berguna dari data transaksi skala besar. Menurut Agrawal (1996) *Association rules* merupakan sebuah ekspresi $X \Rightarrow Y$ yang mana X berkemungkinan mengandung Y . X adalah *antecedent* atau bisa disebut penyebab dan Y adalah *consequent* atau disebut akibat. Isi dari X dan Y adalah sebuah atau kumpulan beberapa barang. Menurut Garg (2018) ada beberapa ukuran dan perhitungan didalam teknik *association rules*, berikut penjelasannya:

1. Support

Ukuran ini memberitahu seberapa sering sebuah *item* ada didalam semua transaksi. Berikut perhitungan *support* di persamaan 2.1:

$$\text{Support}(\{X\} \rightarrow \{Y\}) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Total number of transactions}} \dots (2.1)$$

2. Confidence

Ukuran ini memberitahu kemungkinan kemunculan *item* Y yang biasa disebut *consequent* terkandung di *item* X yang biasa disebut *antecedent*. Berikut perhitungan *confidence* di persamaan 2.2:

$$\text{Confidence}(\{X\} \rightarrow \{Y\}) = \frac{\text{Transactions containing both } X \text{ and } Y}{\text{Transactions containing } X} \dots (2.2)$$

3. *Lift*

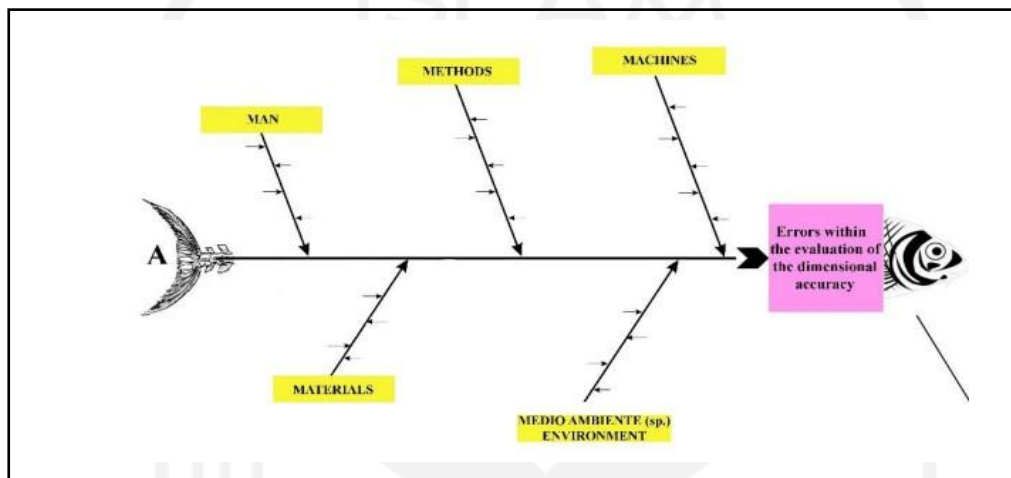
Ukuran yang mengontrol *support* (frekuensi) di *consequent* atau *Y* disaat yang bersamaan pula menghitung probabilitas bersyarat untuk kemunculan *Y* dibawah kondisi *X* atau dengan kata lain *Y* kemungkinan akan muncul jika *X* muncul. berikut perhitungan *Lift* di persamaan 2.3:

$$Lift(\{X\} \rightarrow \{Y\}) = \frac{(Transactions\ containing\ both\ X\ and\ Y)/(Transactions\ containing\ X)}{Fraction\ of\ transactions\ containing\ Y} \dots (2.3)$$

Hasil dari perhitungan *lift* akan mendapatkan beberapa kemungkinan diantara lain hasil nilai perhitungan kurang dari angka 1 dianggap korelasi antar dua variabel negatif, hasil sama dengan angka 1 dianggap dua variabel berdiri sendiri atau tidak ada korelasi, dan hasil yang lebih dari angka 1 mempunyai korelasi positif antar dua variabel (Hussein, 2015). Kemudian menurut Lutkevich (2017) *association rules* menggunakan model *machine learning* untuk menganalisa pola data, atau kejadian bersama antar data didalam *database*.

2.1.8 *Fishbone diagram*

Menurut Liliana (2016) *Fishbone diagram* didefinisikan sebagai representasi grafik yang secara sistematis mengilustrasikan hubungan antara hasil dan penyebabnya. Instrumen grafik yang terdapat di *fishbone diagram* ditujukan untuk memahami penyebab yang membuat cacat pada produk dan instrumen grafik digunakan untuk menganalisa hubungan antara masalah dan semua potensi penyebab. Berikut gambar untuk diagram *fishbone*:



Gambar 8 *Fishbone diagram*

Sumber : Liliana (2016)

Pada gambar 8, terdapat 5 sumber penyebab terjadinya masalah, yang pertama adalah *Man*, yang mana manusia sebagai penyebab masalah. Sebagai contoh adalah kelelahan, kurang pengalaman, ceroboh dan lain-lain. Yang kedua adalah *Methods*, kesalahan akibat ketidakcocokan dan penerapan suatu metode. Contohnya adalah kesalahan dalam pengukuran, penggunaan alat, salah prosedur dan lain-lain. Ketiga adalah *Machine*, yang mana ketidakmampuan mesin dalam menerapkan suatu metode, sehingga menyebabkan masalah baru. Contoh tidak presisinya suatu mesin dalam pengukuran suatu produk. Keempat adalah *Materials* bahan yang tidak sesuai dengan standar produk, sehingga muncul masalah terkait *quality product*. Kelima adalah *Environment*, yang mana lingkungan berkemungkinan menjadi penyebab masalah suatu produk. Contoh adalah temperatur yang terlalu panas, sehingga mempengaruhi *quality product*.

2.2 Kajian Induktif

Kajian Induktif berisi penelitian terdahulu yang memiliki beberapa metode yang sama dengan penelitian ini.

Aliguliyev et al (2015) melakukan penelitian tentang pembuatan sebuah sistem deteksi artikel yang bermuatan unsur terorisme. Penelitian bertujuan mengangani artikel terorisme yang mana artikel tersebut menyamar sebagai situs terpercaya yang ada di internet seperti situs pemerintah atau *e-government*. Metode yang digunakan yaitu metode yang berdasarkan teknik *text mining*. Metode ini dipilih karena data yang didapat mayoritas berupa teks. Hasil penelitian didapatkan metode dapat digunakan, namun ada beberapa kendala dalam teknik *text mining* Ramiz-Gunay yang mana beberapa kendala ini dapat mempengaruhi keakuratan metode yang digunakan.

Metsker et al (2018) melakukan penelitian mengenai pengembangan *e-government* dalam bidang hukum. Tujuan penelitian ingin mengetahui hubungan dan struktur proses penerapan undang-undang dan untuk mengembangkan dasar rekomendasi untuk memperbaiki hukum dengan penggunaan metode pengolahan data multidimensi. Metode *text mining* digunakan dalam penelitian ini dikarenakan data yang digunakan tidak berstruktur. Hasil yang didapat merupakan saran penggunaan *decision tree* karena salah satu cara yang paling mudah diaplikasikan dan telah dapat mencapai tujuan penelitian yang ada.

Payton et al (2018) melakukan penelitian mengenai laporan tentang masalah kesehatan mental yang dialami para mahasiswa/i. tujuan dari penelitian ini yaitu memberikan peningkatan atau perbaikan terhadap jasa kesehatan mental yang memberikan dukungan dan pelayanan kepada mahasiswa/I yang mengalami kesulitan. Metode *text mining* digunakan karena data yang didapat mayoritas berupa teks tidak berstruktur. Alat bantu yang dipakai bernama SAS Text Miner untuk menganalisa 165 referensi yang dipublikasikan dari rentang waktu 2010 sampai 2015 dan berisi tentang kesehatan mental. Hasil penelitian menunjukkan terdapat 6 faktor yang membuat para mahasiswa/i mengalami sakit mental yaitu *age, race, crime, services, aftermath, violence and sexual assault*. Dari keenam faktor tersebut faktor *services* menempati urutan

pertama dengan kasus pelayanan terjadi sampai pada 113 kasus dari 165 kasus, sedangkan faktor *race* menempati urutan kedua dengan kasus ras terjadi sampai pada 30 kasus dari 165 kasus. Dari ke 6 faktor yang didapat kemudian terbentuklah sebuah usulan seperti melakukan penanggulangan *preventif* dan *proactive* yang berfokus pada ke 6 faktor tersebut.

Wijaya et al (2016) membuat penelitian berupa klasifikasi dokumen untuk identifikasi konten atau isi dari *e-government*. Tujuan penelitian yaitu untuk mengetahui konten politik dan ekonomi yang ada di *e-government* serta ingin menjadikan situs *e-government* menjadi lebih informatif. *Text mining* dijadikan sebuah metode karena banyaknya data yang berupa teks dan karena sudah terbukti sukses *output* yang dihasilkan, sedangkan untuk teknik klasifikasi digunakan klasifikasi naïve bayes karena berkualitas dan waktu *computing* yang singkat. 260 dokumen politik dan 222 dokumen ekonomi digunakan sebagai data *training* dan 40 dokumen dijadikan data *testing*. Hasil yang keluar adalah akurasi dalam menentukan isi dari situs *e-government* yang berupa politik atau ekonomi. Politik mendapatkan akurasi sebesar 95 %, sedangkan ekonomi sebesar 68,75 %. Rata-rata akurasi keseluruhan sebesar 85%.

Zhou et al (2019) melakukan penelitian mengenai identifikasi masalah atau keinginan pengguna aplikasi perangkat seluler m-library. Tujuan penelitian untuk mengetahui semua masalah atau keinginan yang dibutuhkan oleh pengguna aplikasi m-library dan menemukan masalah atau keinginan yang paling penting berdasarkan data dari pengguna. Penelitian ini menggunakan metode *text mining* untuk bisa mengambil informasi dari data tekstual. Data yang diambil berjumlah 27,000 review dari pengguna aplikasi m-library yang tersebar dari 16 provinsi dan daerah otonomi china. Hasil yang keluar berupa tiga tipe perbaikan: fungsi, teknologi dan pengalaman. Ketiga tipe itu bisa dibagi lagi menjadi 6 sub tipe: banyaknya fungsi yang dapat dinikmati, kelayakan fungsi, kemudahan teknologi, stabilitas teknologi, optimilasi pengalaman dan kustomisasi pengalaman.

Lai et al (2017) melakukan penelitian tentang analisa komentar pengguna di aplikasi Google Fit. Tujuan dari penelitian diantara lain masalah yang dialami pengguna, dan hal-hal yang dicari pengguna. Metode *Text mining* dipilih karena banyaknya data berupa teks serta hanya dibutuhkan jumlah atau frekuensi kata dari komentar pelanggan.

Terdapat tiga hasil untuk menjawab tujuan dari penelitian ini di antara lain hasil untuk masalah yang dialami pengguna yakni terdapat masalah kesesuaian antara aplikasi Google Fit dengan perangkat seluler, dan masalah pada koneksi dan data sinkronisasi. Untuk hasil hal-hal yang dicari pengguna yakni berupa catatan tentang aktivitas fisik pengguna seperti jalan trek, jarak, waktu, dan kalori yang dibakar ketika sedang jalan atau jogging.

Cheng et al (2020) melakukan penelitian tentang analisa komentar para pengguna aplikasi perangkat seluler pada domain *digital banking* di aplikasi bank asal Filipina yang bernama CIMB Bank PH. Tujuan dari penelitian ini yakni ingin mengetahui keunggulan dan kelemahan aplikasi CIMB bank PH. Metode yang digunakan yaitu *text mining* karena data yang diambil dari Google Playstore berupa teks tidak berstruktur. Selain metode *text mining* terdapat teknik untuk membantu menjalankan *text mining* yaitu teknik *association rules* untuk mengetahui korelasi antar kata yang memiliki jumlah atau frekuensi terbanyak. Hasil didapatkan keunggulan dan kelemahan aplikasi CIMB Bank PH. Untuk keunggulan aplikasi CIMB Bank PH penarikan uang bisa dilakukan yang mana saja, sedangkan kelemahan berupa proses registrasi atau verifikasi yang bank tawarkan perlu diperbaiki hal ini berdasarkan dari korelasi antar kata dari “deposit”, “transfer”, dan “problem”.

Habib et al (2018) dalam penelitiannya membuat evaluasi ukuran kesuksesan untuk para aplikasi yang bergerak di bidang perdagangan atau perniagaan. Tujuan penelitian ini untuk mencari faktor-faktor yang bisa dijadikan dasar sebagai ukuran kepuasan. Metode *text mining* dan teknik *association rules* digunakan dalam penelitian ini. *Text mining* digunakan karena data yang diambil berupa teks, sedangkan *association rules* digunakan untuk mencari faktor-faktor ukuran kesuksesan menjadi lebih akurat. Data yang diambil sebanyak 7000 komentar dari aplikasi Uber, 1500 dari aplikasi Grab dan 1200 dari aplikasi Lyft. Hasil yang keluar yaitu ada delapan faktor ukuran kesuksesan sebuah aplikasi yang bergerak di bidang perdagangan yakni *system quality*, *information quality*, *process quality*, *service quality*, *use*, *user satisfaction*, *individual benefit*, dan *IT infrastructur*.

Lekham et al (2021) melakukan penelitian terhadap kasus pasien yang tidak datang kerumah sakit setelah membuat jadwal dengan rumah sakit. Tujuan penelitian untuk mengidentifikasi faktor mana yang memiliki pengaruh besar yang membuat pasien tidak dapat datang. Metode yang digunakan berupa *text mining*, *pareto analysis*, dan *fishbone diagram*. *Text mining* digunakan untuk mengumpulkan data, merubah data yang bersifat tidak terstruktur menjadi terstruktur dan mengumpulkan data berupa kata atau kalimat dengan frekuensi kemunculan tinggi. *Pareto analysis* digunakan untuk mengetahui diantara kata atau frasa penyebab terjadinya permasalahan dengan frekuensi kemunculan tinggi vital atau tidak. *Fishbone diagram* digunakan sebagai alat visual dan alat yang lebih dalam untuk analisa mengapa kata atau frasa penyebab tersebut terjadi. Faktor yang didapat terdapat 5 dan frasa “*personal related issues*” menadi frasa tertinggi yang membuat pasien tidak dapat datang kerumah sakit.

Sinha et al (2020) meneliti tentang kualitas jasa di perusahaan telekomunikasi menggunakan ulasan daring dari pelanggan dan pekerja. Tujuan penelitian mencari akar penyebab terjadinya permasalahan di perusahaan telekomunikasi. Metode yang digunakan berupa *text mining* dan *fishbone diagram*. *Text mining* digunakan untuk *text preprocessing*, sedangkan *fishbone diagram* digunakan sebagai alat *visual* untuk mempermudah memahami konsep yang diberikan dan yang kedua sebagai alat untuk mmengetahui penyebab dari permasalahan yang sedang dihadapi. Hasil yang didapatkan ada dua sumber yaitu pelanggan dan pekerja. Dari pelanggan dibagi menjadi dua lagi yaitu positif dan negatif. Untuk positif pelanggan menyukai kecepatan internet yang tinggi dan program telepon, sedangkan data negatif berupa sulitnya mobilitas, *customer service* yang biasa saja, tagihan yang tidak tepat dan koneksi yang buruk di sebagian tempat. Sama seperti sumber pelanggan sumber pekerja dibagi menjadi dua yaitu positif dan negatif. Data positif berupa keuntungan yang dibagi, terdapatnya bonus, dan sebagainya, sedangkan data negatif yaitu upah yang rendah, adanya korupsi ditingkat *management*, pelatihan yang kurang dan waktu antara kerja dan libur tidak seimbang.

Tabel 1 memperlihatkan rangkuman dari sepuluh penelitian terdahulu yang mana sepuluh penelitian tersebut mencakup metode dan teknik yang ada di penelitian ini.

Tabel 1 Penelitian Terdahulu

No	Penulis (Tahun)	Judul	Metode & teknik	Objek Penelitian
1	Aliguliyev et al (2015)	<i>Detecting Terrorism-Related Articles on The E- Government Using Text-Mining Techniques</i>	<i>Text mining, Klasifikasi SVM, Naïve Bayes, Ramiz Gunay</i>	Penanganan terror di <i>e-government</i>
2	Metsker et al (2018)	<i>Text and Data Mining Techniques in Judgment Open Data Analysis for Administrative Practice Contol</i>	<i>Text Mining, Machine Learning</i>	Pengembangan kinerja badan hukum, dan <i>e- government.</i>
3	Payton et al (2018)	<i>Text Mining Mental Health Reports for Issues Impacting Today's College</i>	<i>Text mining</i>	Perbaikan dan pengembangan jasa kesehatan mental

No	Penulis (Tahun)	Judul	Metode & teknik	Objek Penelitian
		<i>Students: Qualitative Study</i>		
4	Wijaya (2016)	Naïve Bayes <i>Classification</i> pada Klasifikasi Dokumen Untuk Identifikasi Konten <i>E-Government</i>	<i>Text mining,</i> klasifikasi Naïve Bayes	Identifikasi konten <i>e-government</i>
5	Zhou (2019)	<i>User-opinion mining for mobile library apps in china: exploring user improvement needs</i>	<i>Text mining</i>	Identifikasi masalah atau keinginan pengguna aplikasi m-library
6	Lai et al (2017)	<i>Analysis of User Feedback in The Mobile App Store Using Text Mining: A Case Study of Google Fit</i>	<i>Text mining</i>	Analisa komentar pengguna di aplikasi Google Fit
7	Cheng et al (2020)	<i>Analysing Digital Banking Ulasan Using Text Mining</i>	<i>Text mining,</i> <i>association</i> <i>rules</i>	Analisa pengguna aplikasi <i>digital</i>

No	Penulis (Tahun)	Judul	Metode & teknik	Objek Penelitian
				banking CIMB Bank PH
8	Habib et al (2020)	<i>Succes Measures Evaluation for Mobile Commerce Using Text Mining based on Customer Tweets</i>	<i>Text Mining, Association Rules</i>	Evaluasi ukuran kesuksesan
9	Lekham et al (2021)	<i>A Root Cause Analysis Using Text Mining for Patient No-Shows : A Case Study of Primary Care Centers Serving Rural Areas</i>	<i>Text mining, pareto analysis , dan fishbone diagram</i>	Identifikasi penyebab terjadinya pasien yang tidak datang ke rumah sakit setelah dijadwalkan.
10	Sinha et al (2020)	<i>Improving the service quality of telecommunication companies using online customer and employee ulasan analysis</i>	<i>Text mining, dan fishbone diagram</i>	Identifikasi Penyebab masalah pada kualitas jasa di perusahaan telekomunikasi

BAB 3

METODE PENELITIAN

3.1 Jenis Data

Data terdapat dua jenis yaitu data primer dan sekunder. Data primer yaitu data yang pengambilannya secara langsung di lapangan, sedangkan data sekunder didapatkan dari informasi yang sudah ada. Data primer didapatkan dengan metode *web scrapping* menggunakan aplikasi AppFollow, sedangkan data sekunder bisa didapatkan melalui buku, artikel, jurnal, dan berbagai situs yang berkaitan dengan informasi yang dibutuhkan. Dalam penelitian ini jenis data yang digunakan adalah data Primer yang diambil sejak aplikasi dibuat hingga mulai pengoahan data. Data diperoleh dari aplikasi BPJSTKU yang dapat diunduh Google Playstore.

3.2 Prosedur Penelitian

3.2.1 Tahap Persiapan Data

Sebelum masuk untuk mengumpulkan data, diperlukan persiapan dahulu seperti menentukan data yang akan diambil, tempat pengambilan data, alat yang digunakan untuk mengambil data, dan alat yang digunakan untuk memproses data agar mendapatkan hasil. Data yang diambil berupa data teks dari komentar para pengguna aplikasi BPJSTKU. Pengambilan data dilakukan di situs web bernama Google Play karena jumlah pengguna yang banyak, rating yang kurang bagus, dan masih banyak pengguna yang aktif melakukan komentar atau ulasan aplikasi BPJSTKU. Alat yang digunakan untuk mengambil data atau data *scraping* adalah *Software* AppFollow digunakan karena mudahnya penggunaan *software* tersebut dalam pengambilan data. Alat bantu untuk memproses data adalah Python 3.8.1 dan Microsoft excel 2016. Python digunakan karena terkenal akan mudahnya dalam menggunakan *software* tersebut dan merupakan salah satu *software* yang populer dalam dunia analisa data, sedangkan untuk microsoft excel 2016

digunakan sebagai alat pemastian bahwa data yang diproses oleh python sesuai dengan keinginan.

3.2.2 Tahap Pengumpulan Data

Pengumpulan data dimulai dari penggunaan *software* AppFollow di situs web AppFollow. Pertama, setelah masuk di situs web AppFollow pilih pilihan “*get start for free*” dari dua pilihan yang ditawarkan di menu utama, kemudian isi formulir dengan mengisi *email*, nama, perusahaan dan pekerjaan. Kedua, pilih aplikasi yang diinginkan. Ketiga, pilih “*Reply to Review*” di menu *Rating & Review*. Keempat, atur tanggal komentar pengguna menjadi “*All Time*” dan kemudian tekan pilihan “*export*”. Kelima, pilih unduh “*All Review (using current filters)*” dalam bentuk CSV. Untuk penampilan gambar bisa dilihat dibagian lampiran dengan urutan pertama yaitu gambar L.1, kedua dengan gambar L.2, ketiga dengan gambar L.3, keempat dengan gambar L.4, dan kelima dengan gambar L.5.

3.2.3 Tahap Pengolahan Data

Tahap pengolahan data menggunakan *software* python 3.8.1 untuk memproses data primer yang didapat, kemudian dilanjutkan ke metode *fishbone diagram* untuk mengetahui penyebab terjadinya masalah yang dialami oleh aplikasi BPJSTKU. Proses *text mining* yang dipakai di penelitian ini yakni *text preprocessing*, dan *data mining*. *Text normalization* dan *tokenization* dari *text preprocessing* akan dipakai. Untuk teknik yang digunakan dalam proses *data mining* yaitu *association rules*. Dalam membantu untuk melanjutkan ke metode berikutnya yaitu *fishbone diagram* maka digunakan desain visual seperti *wordcloud*.

Penelitian dimulai dari memasukkan data ke *software* python melalui *interface* Jupyter Notebook. Data yang dimasukkan berupa komentar dan peringkat para pengguna. Komentar pengguna sebagai data utama yang akan diolah, sedangkan peringkat sebagai penentu sentimen positif atau negatif komentar itu. peringkat dengan < 3 dijadikan negatif, sedangkan > 3 dijadikan positif. Penggunaan peringkat sebagai penentu atau

label menjadikan penelitian ini *supervised* dikarenakan sentimen pengguna sudah ditentukan pengguna sendiri saat membuat ulasan di aplikasi Google Playstore.

Setelah memasukkan data, lanjut pada proses *pre processing* dibagi menjadi dua proses yang akan dipakai yaitu proses *text normalization* dan *tokenization*. Pada tahap ini diperlukan modul dari python yakni Pandas dan NLTK. Modul digunakan untuk mempercepat efisiensi dalam melakukan *text mining*. *Text normalization* yaitu proses membuat data mentah menjadi data dengan format yang diinginkan dengan modul pandas seperti: membuat semua huruf menjadi kecil atau kapital, pengilangan data yang tidak berupa teks seperti emoji, dan penghilangan tanda baca. Sesudah melakukan normalisasi pada teks komentar dilanjutkan membuat label berdasarkan *Peringkat* dengan *Peringkat* 1 dan 2 berarti negatif dan 4 dan 5 positif.

Data yang sudah melalui proses *text normalization*, dilanjutkan pada proses *tokenization*. Proses *tokenization* berfungsi merubah kalimat menjadi kumpulan kata. Setelah menjadi kumpulan kata, data dimasukkan ke program *wordcloud* dengan bantuan modul Matplotlib dan NLTK. Modul Matplotlib digunakan untuk memunculkan visualisasi *wordcloud* yang mana seperti kumpulan huruf acak, dan kata ukuran yang paling besar menandakan bahwa kata tersebut memiliki frekuensi tinggi atau jumlah kata tersebut di data banyak, akan tetapi setelah kata bermunculan di *wordcloud* terdapat beberapa kata tidak berarti seperti: “saya”, “kamu”, “dia”, “.” Dan lain-lain. Untuk menghilangkan kata tidak berarti modul NLTK digunakan untuk membuat *stopword* yang berfungsi menghilangkan kata yang tidak berarti tersebut, namun modul NLTK belum cukup dalam menghilangkan semua kata yang tidak berarti karena keterbatasan kata di *database* modul NLTK, oleh karena itu dibuatlah *stopword* manual yang digabung dengan *stopword* NLTK. Setelah *stopword* dilakukan, kata-kata berarti mulai bermunculan, akan tetapi terdapat permasalahan baru lagi dengan adanya dua kata berarti sama namun penulisan satu diantara dua kata tersebut tidak lengkap, hal ini mengakibatkan salah perhitungan frekuensi kemunculan dalam sebuah kata. Untuk menyelesaikan tersebut dibuatkanlah koding manual yang bisa dilihat pada gambar A.9.

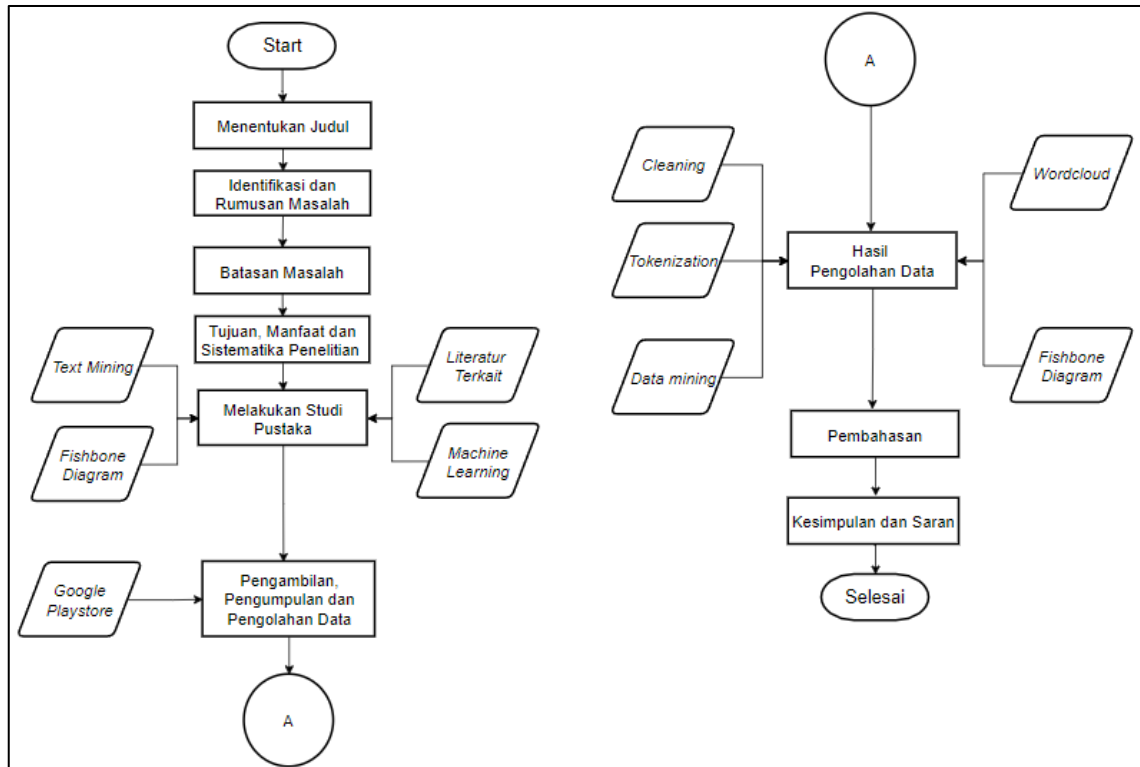
Pada tahap selanjutnya yaitu proses data *mining*. *association rules* berfungsi untuk membuat asosiasi antar kata dan kemudian menemukan nilai korelasi antar kata. Proses

ini memerlukan modul Mlxtend untuk membantu efisiensi koding. Sebelum menggunakan modul Mlxtend data diubah dulu yang semula berbentuk dataframe menjadi kumpulan kalimat yang sudah diberi *stopword*, kemudian menghilangkan kolom *Peringkat* dan label sentimen. Data yang sudah menjadi kumpulan kalimat ini dibuat menjadi daftar multidimensi, lalu modul Mlxtend baru bisa masuk. Data yang sudah berbentuk daftar multidimensi diubah kata setiap kata menjadi “*true*” dan “*false*”. Setelah kata berubah *true* dan *false* maka berlanjut untuk mengeluarkan ukuran dan perhitungan dari proses *asscociation rules* yaitu *support*, *confidence* dan *lift*. Data yang berfrekuensi tinggi dan memiliki korelasi yang kuat dengan data lain akan dipilih lalu diidentifikasi secara deskripsi.

Setelah menyelesaikan proses *asscociation rules* maka dilanjutkan ke proses *fishbone diagram*. *Fishbone diagram* berfungsi sebagai alat visual dan analisa masalah dan penyebab. Hasil analisa masalah dan penyebab didapatkan dari *wordcloud* dan *association rules*. Pemilihan hasil berdasarkan jumlah kata yang memiliki frekuensi terbesar. Diketahui terdapat lima kata yang akan dipilih dalam penelitian ini. Penggunaan metode *fishbone diagram* mempermudah penelitian untuk mengetahui dengan jelas penyebab permasalahan yang kemudian hasil penelitian dapat disimpulkan dengan mudah.

3.3 Diagram Alur

Diagram alur menggambarkan secara garis besar penelitian ini dilakukan. Berikut gambar tersebut:

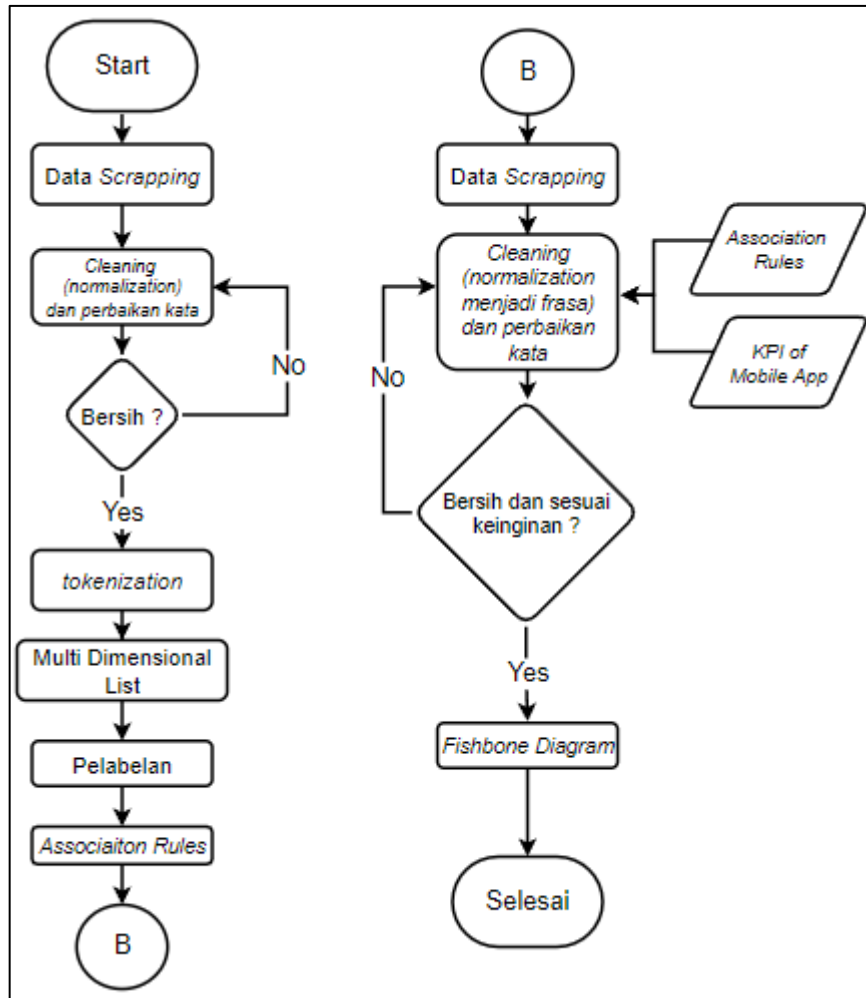


Gambar 9 Diagram Alur

Terlihat pada gambar 9 penelitian dimulai dari penentuan judul kemudian masuk ke bab 1 yang dimulai dari identifikasi dan rumusan masalah pada objek yang diteliti. Kemudian memberikan batasan untuk menyelesaikan masalah yang ada di objek. Setelah itu memaparkan tujuan, manfaat dan sistematika penelitian yang kemudian masuk ke bab 2 pada tahap melakukan studi pustaka dengan mengumpulkan data dari literatur terkait, *machine learning*, *text mining*, dan *fishbone diagram*. Setelah itu masuk ke bab 3 dengan tahap pengambilan, pengumpulan dan pengolahan data yang mana untuk tahap pengambilan dan pengumpulan dibantu dengan tambahan informasi yang didapat dari Google Playstore dengan bantuan alat AppFollow. Setelah itu masuk ke bab 4 pada tahap hasil pengolahan data yang mana memerlukan data koding pemrograman dan cara menggunakannya. Data koding dan pemrograman meliputi *cleaning*, *tokeization*, *data mining*, dan *wordcloud*. Untuk bagian *fishbone diagram* data yang diambil adalah

template diagram *fishbone*. Setelah mendapatkan data yang dibutuhkan penelitian dapat dilakukan. Dari hasil yang didapat kemudian masuk ke bab 5 pada tahap pembahasan yang mana tujuan pada tahap ini untuk menjelaskan hasil dari pengolahan data dari bab 4. Setelah melakukan pembahasan lalu masuk ke bab 6 yaitu bab terakhir di penelitian ini dengan melakukan tahap kesimpulan dan saran yang bertujuan untuk memberi kesimpulan dan saran dari tahap pembahasan yang ada di bab 5. Dari diagram alur garis besar tersebut terdapat bagian yang diperlukan diagram alur sendiri guna untuk lebih mudah memahami penelitian ini. Bagian tersebut adalah pada bab 4 tahap hasil dan pengolahan data. Berikut diagram alur untuk tahap hasil dan pengolahan data:





Gambar 10 Diagram Alur Hasil dan Pengolahan Data

Gambar 10 merupakan diagram alur tahap hasil dan pengolahan data. Tahapan dimulai dari mengumpulkan data dari Google Playstore dengan bantuan alat AppFollow. Kemudian melakukan *cleaning* dan perbaikan kata. Jika *cleaning* dan perbaikan kata sudah bersih dan sesuai keinginan maka dilanjutkan pada proses *tokenization* yang mana daftar kalimat dijadikan daftar kata, hal ini dilakukan agar bisa masuk ke proses selanjutnya. Setelah itu data yang sudah di *tokenization* akan dimasukkan ke *association rules*, namun untuk bisa masuk ke proses *association rules* data terlebih dahulu dijadikan daftar multi dimensi dari daftar kata dan kemudian di beri label angka antara 0 dan 1. Setelah pelabelan selesai data, baru bisa dimasukkan ke dalam proses *association rules*. Hasil dari *association rules* ini kemudian berfungsi sebagai petunjuk untuk mengisi *fishbone diagram*. Untuk mengisi *fishbone diagram* diperlukan data asli dari proses *data scrapping*. Kemudian melakukan *cleaning* dan perbaikan kata yang mana kumpulan kata

didapat dari tahap *data scrapping* dan kemudian dipilih kata yang diperlukan dengan petunjuk dari *association rules* dan kategori didalam *key performance indicator of mobile application*, lalu kumpulan kata tersebut dijadikan sebuah frasa yang diinginkan. *Key performance indicator of mobile application* hanya bisa di aplikasikan setelah melakukan *association rules*, karena proses *cleaning* dan perbaikan kata menjadi frasa membuat algoritma *association rules* susah membaca frasa dan pada akhirnya tidak dapat terbaca. Diketahui bahwa *association rules* perlu dua kata identik di kata *antecedent* atau *consequent* untuk dapat terbaca. Jika salah satu kata dari dua kata identik tadi tidak ada maka algoritma *association rules* tidak dapat terbaca. Contoh algoritma dapat menangkap sebuah kata atau frasa: “aplikasi gagal aplikasi” dan “aplikasi error login error”. Dua kata identik dari contoh yakni aplikasi dan error. *Association rules* tidak dapat membaca frasa seperti: “aplikasi_bpjstku_gagal login_gagal”, dua frasa dalam satu baris tersebut tidak dapat dibaca oleh algoritma *association rules*, jika ingin terbaca satu baris memerlukan dua frasa identik dan satu frasa, contoh seperti: “aplikasi_bpjstku_gagal login_gagal aplikasi_bpjstku_gagal”. Frasa-frasa yang didapatkan pada proses sebelumnya kemudian dimasukkan kedalam *fishbone diagram*.

BAB 4

PENGOLAHAN DATA

4.1 Text Mining

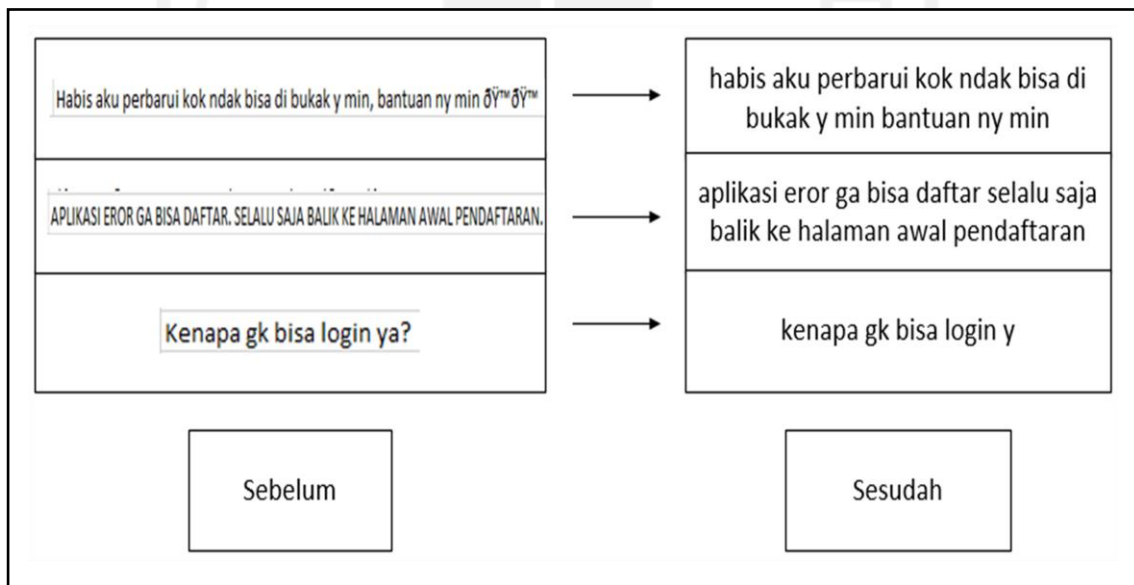
Metode *text mining* digunakan karena ingin mendapatkan pola menarik di data yang besar dan berupa tekstual. Terdapat dua proses utama berupa *text preprocessing* dan data *mining* serta dua tambahan proses didalam proses *text preprocessing* yaitu *text normalization* dan *tokenization*. Proses dilakukan pertama kali melalui *text preprocessing* dan kedua proses tambahan didalamnya, lalu dilanjutkan pada proses data *mining*. Data yang ditampilkan merupakan data sebelum dan sesudah diolah. Pengolahan data menggunakan *software* python 3.8 dan excell.

4.1.1 Text Preprocessing

Text Preprocessing dimulai dengan memasukkan data mentah ke dalam *software* python melalui *interface* Jupyter Notebook. Memasukkan atau *import* data ini dilakukan dengan bantuan modul *pandas*. Data yang digunakan berupa data ulasan dan peringkat. Proses pembersihan dan pemecahan kalimat menjadi kata akan dilakukan pada tahap ini. Berikut 2 proses yakni proses pembersihan dan pemecahan kalimat menjadi kata :

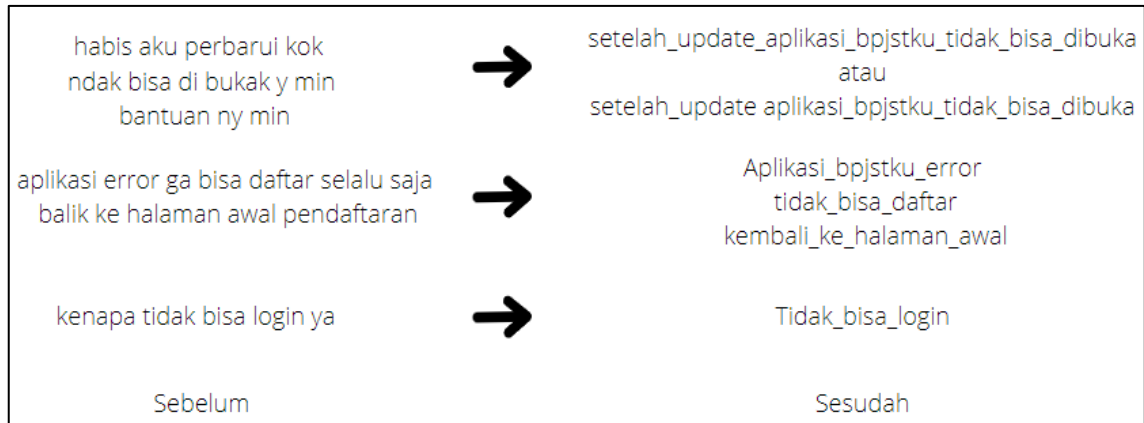
4.1.1.1 Text Normalization

Proses normalisasi teks menjadikan data yang belum mempunyai format atau data masih tercampur dengan kata yang tidak bisa diubah dibaca atau mengganggu menjadi data yang memiliki standart format. Sebelum masuk ke proses, data terlebih dulu dipisah antara data rating negatif dan positif. Data yang digunakan di penelitian ini adalah data negatif, setelah terpilih lalu bisa dilanjutkan ke proses *text normalization*. Hasil dari normalisasi yaitu berupa penghilangan tanda baca dan emoji, dan menjadikan semua font huruf kecil. Contoh normalisasi teks dengan menggunakan python dalam gambar berikut:



Gambar 11 Data mentah ke *Text Normalization* Kata

Pada gambar diatas terdapat dua bagian bernama sebelum dan sesudah. Masing – masing bagian memiliki tiga baris yang memiliki isi satu kalimat. Bagian sebelum merupakan contoh kalimat yang belum dinormalisasi, sedangkan bagian sesudah sudah dinormalisasi. Normalisasi untuk baris pertama yaitu, kalimat dibuat huruf kecil dan menghilangkan huruf yang tidak dikenal. Normalisasi baris kedua berupa membuat semua huruf menjadi kecil dan menghilangkan tanda titik. Normalisasi pada baris ketiga berupa membuat huruf kecil an menghilangkan tanda koma.



Gambar 12 Data Mentah ke *Text Normalisasi Frasa*

Pada gambar 12 penggunaan normalisasi frasa berlaku setelah melakukan *association rules* yang kemudian hasil frasa dengan petunjuk dari hasil *association rules* dan *key performance indicator of mobile app* akan dimasukkan ke dalam diagram *fishbone*

4.1.1.2 *Tokenization*

Tokenization merupakan proses selanjutnya dari proses *text normalization* di penelitian ini. *Tokenization* dilakukan untuk memecah kalimat menjadi kata-kata. Kata-kata ini kemudian diolah untuk bisa mendapatkan nilai atau *value*. Berikut gambar contoh *tokenization* :

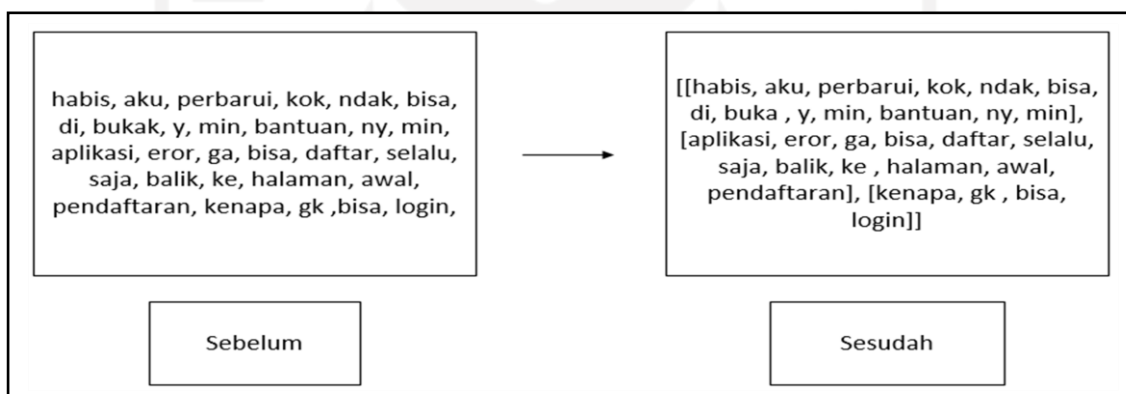


Gambar 13 *Text normalization* ke *Tokenization*

Terlihat pada gambar diatas bahwa sebelum ditokenisasi kata masih berkalimat dan dipisah oleh baris. Setelah ditokenisasi kalimat dipecah menjadi kata per kata tanpa adanya batasan dari baris. Setelah *tokenization* dilanjutkan maka, proses *wordcloud* dan data *mining* bisa dilakukan.

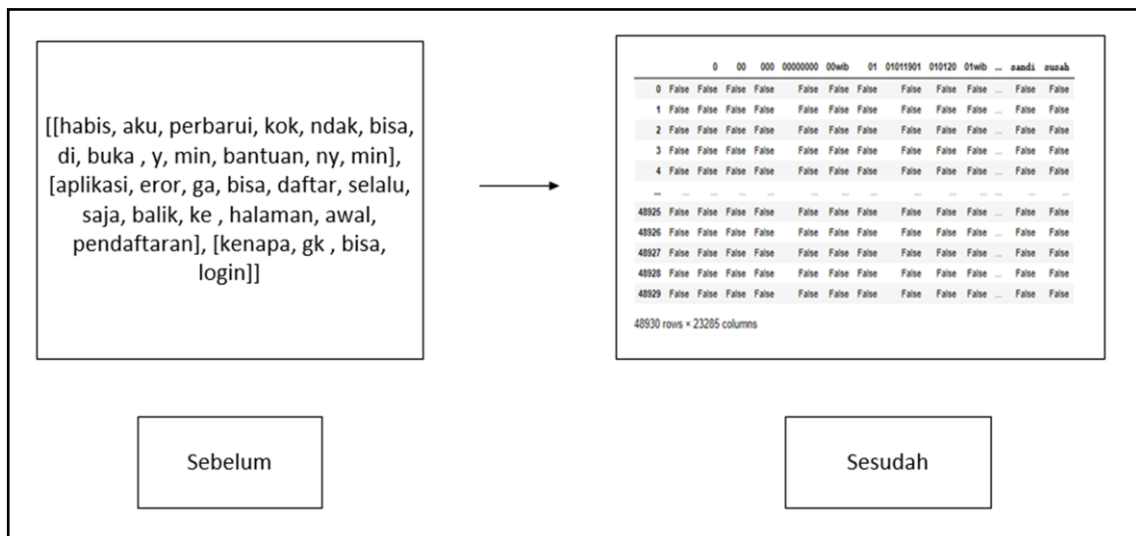
4.1.2 Data Mining

Pada tahap data *mining* digunakan teknik *association rules* untuk mengetahui hubungan antar kata yang sudah dipecah atau *tokenization*. Sebelum masuk ke koding *association rules* data token diubah menjadi data *multidimensional* seperti gambar berikut :



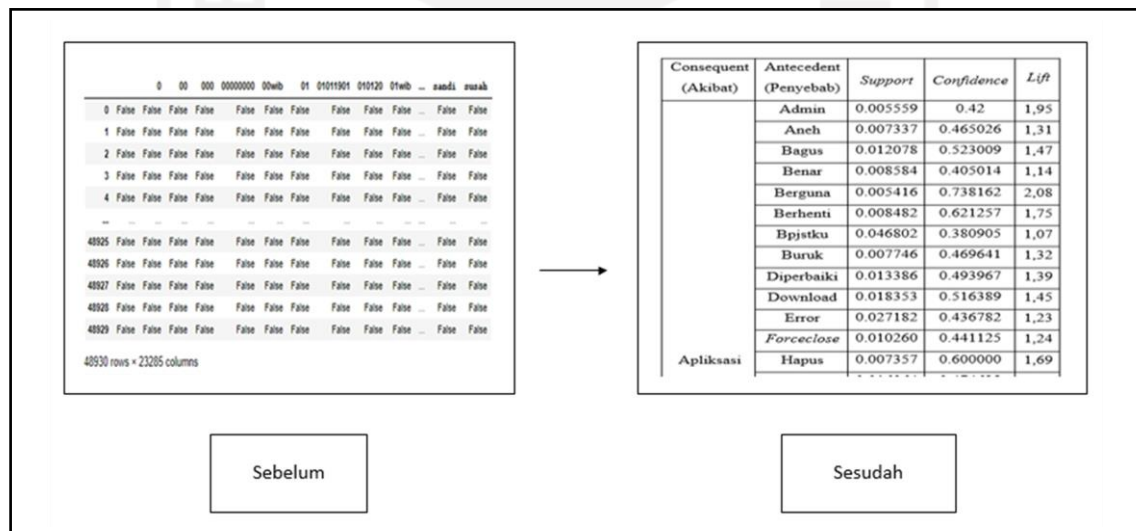
Gambar 14 *Tokenization* ke *Multidimensional list*

Kumpulan kata yang sudah berbentuk *multidimensional* dimasukkan ke proses pelabelan *true* atau *false*. Tahap ini bertujuan untuk mengetahui kata mana yang sama atau berbeda antar baris dan kolom. *True* yaitu untuk label untuk kata sama, sedangkan *false* untuk kata berbeda. Berikut contoh gambar pelabelan:



Gambar 15 Multidimensional list ke Pelabelan true & false

Hasil yang didapat dari pelabelan dilanjutkan ke proses *association rules*. Berikut hasil dari *association rules* :



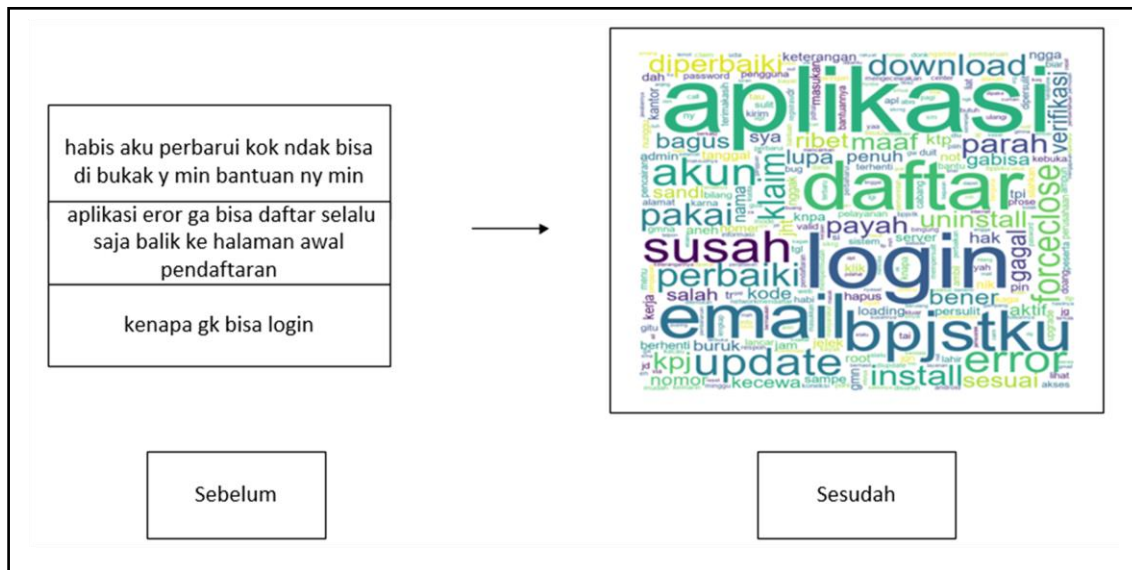
Gambar 16 Pelabelan true & false ke Association Rules

Pada gambar 16 terlihat ada lima *header* yaitu *antecedents*, *consequents*, *support*, *confidence*, dan *lift*. *Antecedents* diartikan sebagai penyebab dari *consequents* secara otomatis *consequent* memiliki arti sebagai akibat. *Support* memiliki arti seberapa sering kata tersebut muncul didalam seluruh frekuensi kata. *Confidence* memiliki fungsi seberapa

besar tingkat kemunculan item *Y* jika mengambil item *X*. Untuk *lift* yaitu untuk mengetahui sudah valid atau tidaknya nilai *confidence* yang didapat. Jika nilai lebih dari 1 maka hubungan antar dua atau lebih kata memiliki hubungan positif, sedangkan untuk nilai kurang dari 1 maka hubungan antar dua atau lebih kata memiliki hubungan negatif. Untuk nilai dekat atau sama dengan 1 maka *X* dan *Y* tidak memiliki hubungan.

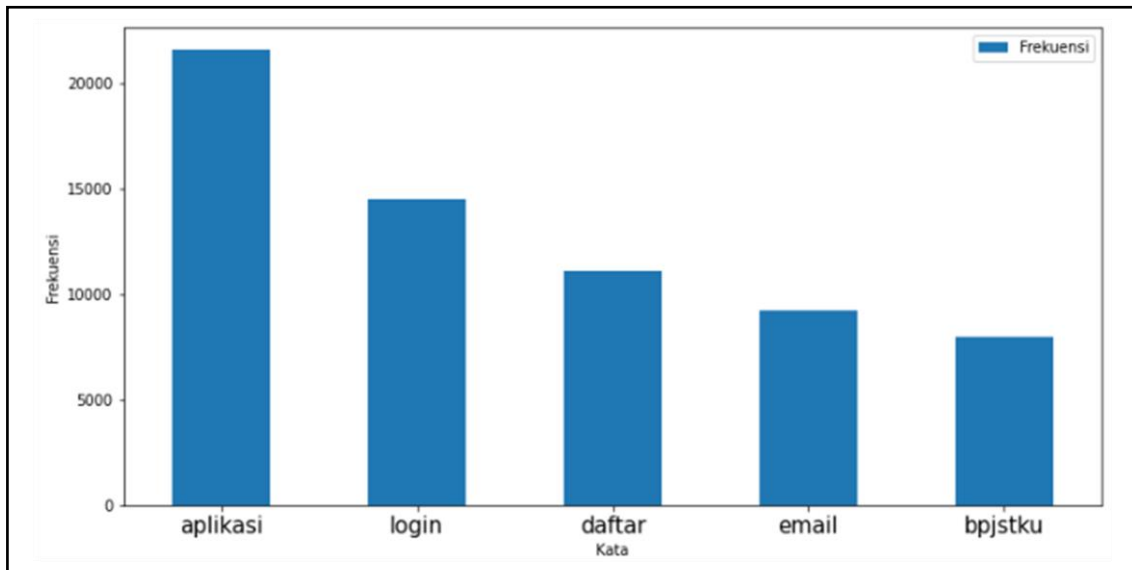
4.2 WordCloud

Pada tahap ini data diolah untuk mencari sejumlah kata dengan frekuensi tinggi yang mana menjadi salah satu dasar untuk menyelesaikan metode *fishbone diagram*. Untuk proses *wordcloud*, data diolah dengan cara divisualisasikan. Data tergambar sebagai kata dengan berbagai ukuran. Kata dengan jumlah frekuensi tinggi direpresentasikan dengan ukuran huruf yang besar di *wordcloud*. Setelah hasil *wordcloud* pertama keluar maka terdapat beberapa kata yang tidak memiliki arti dan kata yang rusak atau memiliki makna sama, maka dari itu diperlukan untuk menghapus kata yang tidak memiliki arti tersebut dengan menggunakan *stopword* manual dan *stopword* dari NLTK indonesia serta memperbaiki kata yang rusak dan kata yang memiliki makna sama dengan koding perbaikan manual. berikut hasil gambar *wordcloud* setelah melalui proses *stopword* dan perbaikan dan penggabungan kata:



Gambar 17 Tokenization ke wordcloud

Untuk mengetahui jumlah frekuensi kata yang memiliki ukuran kata yang besar, maka digunakanlah *bar chart*. Jumlah frekuensi diperlukan, karena ukuran kata yang terdapat di *wordcloud* atau pada gambar 17 hampir memiliki ukuran yang sama. Berikut gambar *bar chart* untuk *wordcloud* :



Gambar 18 Bar chart

Pada gambar 18 diketahui kata “aplikasi” mempunyai jumlah frekuensi tertinggi dengan jumlah sekitar 25.000. untuk urutan kedua, kata”login” mempunyai jumlah frekuensi sekitar 14.000. urutan ketiga, yaitu kata “ daftar” mempunyai jumlah frekuensi sekitar 12.000. urutan keempat dimiliki oleh kata “email” dengan jumlah frekuensi sekitar 9000. Untuk urutan kelima, dimiliki oleh kata “bpjstku” dengan jumlah frekuensi sekitar 8000.

4.3 Fishbone Diagram

Dari hasil yang didapat pada proses sebelumnya yaitu hasil visual dari *wordcloud* dan hasil hubungan dari *association rules*, maka bisa dilanjutkan pada metode selanjutnya yaitu *fishbone diagram*. Untuk hasil dari *wordcloud* didapatkan lima kata berpengaruh yaitu aplikasi, login, email, daftar, dan bpjstku, sedangkan hasil *association rules* menghasilkan hubungan antar kata. Agar mempermudah pembuatan *fishbone diagram* hasil dari kedua proses tersebut akan dijadikan satu dalam bentuk tabel. Berikut contoh tabel tersebut:

Tabel 2 Hasil *association rules* untuk 5 kata

Consequent (Akibat)	Antecedent (Penyebab)	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
Aplikasi	Berguna	0.005	0.73	2,08
	Terhenti	0.008	0.72	2,04
	Berhenti	0.008	0.62	1,75
	Hapus	0.007	0.60	1,69
	Jelek	0.007	0.55	1,55
	Bagus	0.012	0.52	1,47
	Root	0.006	0.52	1,47
	Download	0.018	0.51	1,45
	Pakai	0.018	0.49	1,43
	Diperbaiki	0.013	0.49	1,39
	Install	0.016	0.47	1,34
	Buruk	0.007	0.46	1,32
	Aneh	0.007	0.46	1,31
	<i>Forceclose</i>	0.010	0.44	1,24
	<i>Error</i>	0.027	0.43	1,23
	Parah	0.012	0.42	1,21
	Kecewa	0.008	0.42	1,19
	Admin	0.005	0.42	1,95
	Payah	0.010	0.41	1,16
	Benar	0.008	0.40	1,14
	Bpjstku	0.046	0.38	1,07
	Keterangan	0.005	0.37	1,06
	Salah	0.007	0.36	1,02
	Loading	0.005	0.36	1,02
	Login	0.079	0.35	1,01

Consequent (Akibat)	Antecedent (Penyebab)	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
Login	Password	0.005	0.55	2,49
	Email	0.058	0.46	2,12
	Akun	0.021	0.39	1,76
	Server	0.006	0.39	1,78
	Keterangan	0.006	0.39	1,78
	Sandi	0.007	0.38	1,71
	Daftar	0.057	0.36	1,66
	Pakai	0.012	0.32	1,48
	Salah	0.006	0.32	1,47
	Benar	0.006	0.31	1,41
	Gagal	0.006	0.31	1,24
	Susah	0.024	0.29	1,35
	Update	0.017	0.29	1,32
	Diperbaiki	0.007	0.29	1,32
	Lupa	0.005	0.29	1,34
	Perbaiki	0.013	0.28	1,28
	Gabisa	0.005	0.28	1,24
	Install	0.008	0.25	1,17
	Payah	0.006	0.25	1,15
	<i>Forceclose</i>	0.005	0.25	1,17
Maaf	0.005	0.25	1,15	
Aplikasi	0.079	0.22	1,02	
Parah	0.006	0.22	1,02	
Uninstall	0.007	0.03	1,35	
Consequent (Akibat)	Antecedent (Penyebab)	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
Daftar	Email	0.066	0.53	3,37
	Nik	0.006	0.49	3,12
	Akun	0.024	0.45	2,88
	Ktp	0.005	0.41	2,63
	Keterangan	0.006	0.39	2,49
	Nomor	0.005	0.34	2,17
	Pakai	0.012	0.33	2,10
	Sesuai	0.006	0.32	2,07
	Penuh	0.005	0.32	2,07
	Sandi	0.005	0.29	1,87
	Login	0.057	0.26	1,66
	Kpj	0.005	0.26	1,70
Maaf	0.006	0.25	1,63	

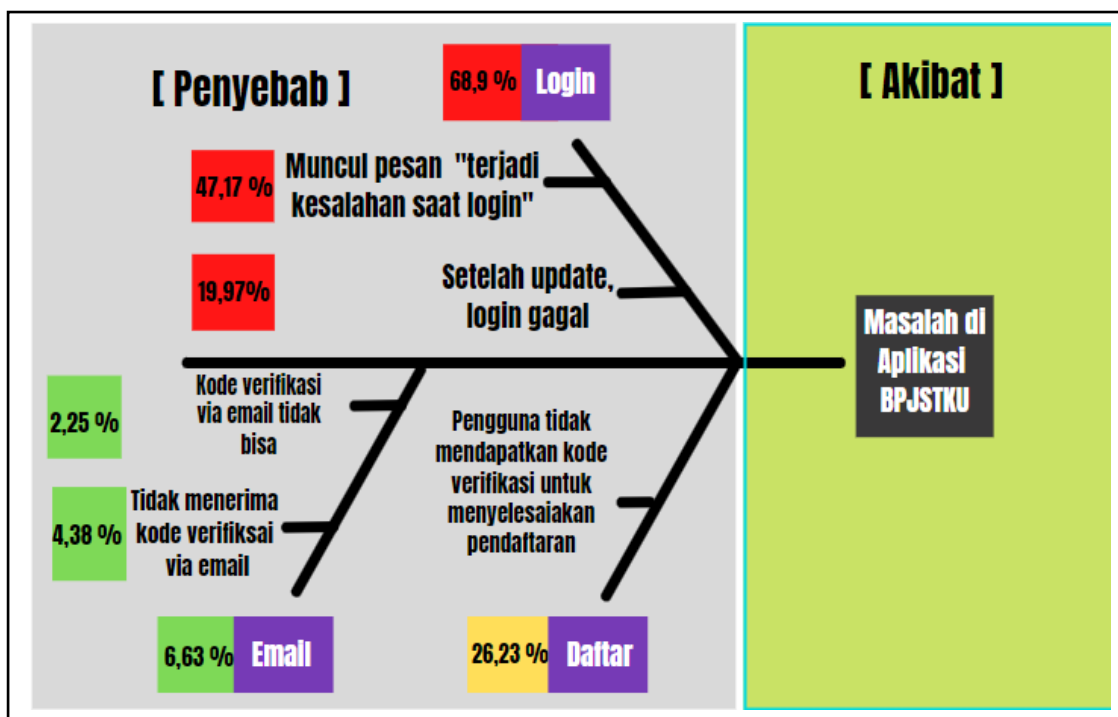
Consequent (Akibat)	Antecedent (Penyebab)	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	Verifikasi	0.005	0.25	1,61
	Salah	0.005	0.25	1,59
	Benar	0.005	0.24	1,58
	Bpjstku	0.027	0.22	1,44
	Susah	0.016	0.19	1,25
	Perbaiki	0.007	0.16	1,05

Consequent (Akibat)	Antecedent (Penyebab)	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	Alamat	0.008	0.84	6,65
	Password	0.006	0.67	5,34
	Sandi	0.010	0.58	4,62
	Lupa	0.011	0.57	4,51
	Verifikasi	0.009	0.47	3,79
	Kode	0.007	0.46	3,73
	Daftar	0.066	0.42	3,45
Email	Akun	0.018	0.35	2,84
	Bener	0.007	0.35	2,83
	Salah	0.006	0.34	2,73
	Pakai	0.012	0.32	2,60
	Maaf	0.005	0.21	1,69
	Bpjstku	0.020	0.16	1,34
	Susah	0.011	0.13	1,07
	Update	0.007	0.13	1,05

Consequent (Akibat)	Antecedent (Penyebab)	<i>Support</i>	<i>Confidence</i>	<i>Lift</i>
	Kantor	0.007	0.53	4,31
	Kerja	0.006	0.45	3,71
	Hak	0.006	0.36	2,96
	Klaim	0.009	0.30	2,45
	Maaf	0.006	0.27	2,27
BPJSTKU	Pakai	0.007	0.19	1,62
	Daftar	0.027	0.17	1,44
	Akun	0.009	0.17	1,40
	Email	0.020	0.16	1,34
	Susah	0.013	0.15	1,28
	Perbaiki	0.006	0.15	1,23
	Aplikasi	0.046	0.13	1,07

Tabel 2 memiliki isi hasil dari *association rules* berupa *antecedent*, *consequent*, *support*, *confidence*, dan *lift*. Dari hasil tersebut kemudian diolah untuk bisa dimasukkan ke *fishbone diagram* dengan cara membandingkan hasil *association rules* dengan ulasan pengguna aplikasi BPJSTKU. Perbandingan ini dilakukan guna mengetahui potensi penyebab lebih detail. Setelah potensi detail penyebab ditemukan kemudian dimasukkan ke diagram *fishbone* untuk mengetahui detail penyebab lebih jelas lagi. Berikut gambar *fishbone diagram* tersebut:





Gambar 19 Fishbone diagram

Dari gambar 19 terlihat hasil olah data yang dimasukkan ke *diagram fishbone*. *Fishbone diagram* memiliki kepala ikan sebagai kepala masalah, yang berisi “Masalah di aplikasi BPJSTKU”. Setelah kepala terdapat tulang belakang yang mempunyai cabang, cabang ini mempunyai kepala yang berarti sebagai penyebab secara garis besar seperti login, daftar, email. Pada setiap cabang kepala penyebab terdapat panah yang mengarah ke batang cabang dengan sebutan sub penyebab. Penyebab – penyebab ini didapat dengan petunjuk dari hasil *association rules* pada tabel 2 dan ulasan para pengguna aplikasi BPJSTKU dengan menggunakan *software python*. Kemudian untuk diketahui bahwa tidak semua hasil dari *association rules* ada di diagram, seperti contoh *consequent* “aplikasi” dan “bpjstku” . kedua *consequent* tidak dimasukkan kedalam diagram bagian penyebab karena kedua *consequent* tersebut masuk kedalam bagian akibat, selain itu kedua *consequent* tersebut hanya memiliki kata sifat yang mana tidak memberi detail kenapa *consequent* tersebut bermasalah, contoh seperti “aplikasi force close”, “aplikasi tidak bisa dibuka”, “bpjstku tidak bisa dibuka” dan lain-lain.

BAB 5

PEMBAHASAN

5.1 Text Mining

Pada *text mining* untuk penelitian ini, ada beberapa tahap proses seperti *text preprocessing*, *Data mining*, dan *Wordcloud*. Tahap *preprocessing* memiliki dua subproses, yakni *text normalization* dan *tokenization*. Untuk data *mining* teknik yang digunakan adalah *association rules* untuk mencari hubungan antar kata. Sebelum masuk ke data *mining*, data yang sudah diolah melewati tahap *tokenization* terlebih dahulu dimasukkan ke dalam proses *wordcloud*. *Wordcloud* digunakan untuk mengetahui frekuensi kata-kata yang diambil dari data yang sudah diolah melalui proses *text preprocessing* dengan tujuan mengetahui kata mana yang berpotensi besar mempengaruhi peringkat aplikasi BPJSTKU.

Pada tahap *preprocessing* dimulai dari proses *text normalization* yang mana menghasilkan data yang bersih atau tidak ada huruf atau kata yang tidak dapat dibaca. Contoh kalimat bisa dilihat pada gambar 9, yang mana tanda hubung, emoji dan huruf yang tidak bisa dibaca dihilangkan. Normalisasi atau pembersihan data dilakukan untuk menghilangkan kata atau huruf yang tidak berarti, sehingga untuk proses *wordcloud* tidak dipenuhi dengan kata-kata tersebut. Kata-kata yang tidak berarti tersebut jika tidak dihilangkan, maka kata-kata yang sesungguhnya memiliki pengaruh akan kalah jumlah dengan kata yang tidak berarti, sehingga akan sulit untuk menemukan kata-kata yang memiliki pengaruh tersebut.

Setelah melakukan olah data di proses *text normalization*, dilanjutkan ke proses selanjutnya yaitu *tokenization*. *Tokenization* bertujuan untuk memecah suatu kalimat menjadi kata-kata. *Tokenization* dilakukan, agar data bisa dilanjutkan ke proses data *mining* dan *wordcloud*. Contoh *tokenization* bisa dilihat pada gambar 10, yang mana

untuk tabel bagian kiri adalah kalimat sebelum *tokenization* dan tabel sebelah kanan sudah *tokenization*.

Setelah proses *tokenization* dan *wordcloud* selesai, dilanjutkan pada proses berikutnya yaitu data *mining*. Teknik yang digunakan untuk proses data *mining* adalah *association rules*, karena data yang dicari atau dibutuhkan adalah data yang bertujuan untuk mengetahui masalah dalam sebuah aplikasi, maka dari itu teknik *association rules* yang memiliki konsep penyebab akibat digunakan. Dari hasil *association rules* terdapat lima kata *consequent* terpilih berdasarkan jumlah kata didalam data dan beberapa kata *antecedent*. Lima kata *consequent* tersebut adalah “ aplikasi ”, “ login ”, “ daftar ”, “ email ”, dan “bpjstku”. Dari daftar kata *consequent* dan *antecedent*, diperoleh perhitungan rumus seperti *support*, *confidence*, dan *lift*. *Support* berarti seberapa sering kedua item muncul dibagi total jumlah data. *Confidence* berarti kemungkinan item *Y* muncul, jika *X* muncul. Kemudian *lift* berarti mengetahui tingkat validasi dari *confidence*. contoh gambar hasil *association rules* bisa dilihat pada tabel 1.

Setelah *association rules* selesai maka dilanjutkan ke proses pengolahan data dari *association rules* menuju *fishbone diagram* menggunakan proses *text normalization* yang mana berfungsi untuk menghilangkan kata yang tidak diperlukan lagi serta merubah kata tidak baku menjadi baku dan yang terakhir yaitu pembuatan frasa agar kalimat lebih mudah dibaca dan tidak terhapus nantinya.

5.2 Wordcloud

Data yang sudah diolah menjadi token, kemudian dimasukkan ke dalam proses *wordcloud* untuk mencari kata berpengaruh. Terlihat pada gambar 14 dan 15 terdapat kata-kata yang memiliki frekuensi paling banyak. Pada gambar 14 ditemukan beberapa kata yang paling menonjol atau secara sekilas dilihat, kata tersebut menjadi perhatian, seperti contoh ukuran kata yang jauh lebih besar dari lainnya. Kata yang paling menonjol berupa “aplikasi”, ”login”, “email”, “daftar”, “bpjstku”. Kelima kata tersebut kemudian dijadikan sebuah prediksi bahwa kelima kata tersebut merupakan penyebab terjadinya kebanyakan masalah pada aplikasi BPJSTKU dengan menggunakan teknik *association rules*.

5.3 *Fishbone diagram*

Fishbone diagram merupakan proses terakhir dari penelitian ini. Hasil yang didapatkan berupa tiga kata *consequent* terpilih yaitu “login”, “email” dan “daftar”. Ketiga kata tersebut dipilih karena memiliki sub penyebab dari kepala penyebab serta berdasarkan petunjuk dari *key performance indicator of mobile application* yang mana permasalahan pada aplikasi di telepon seluler secara umum terdiri dari empat yakni “*app crash*”, “*latency*”, “*loading*”, dan “*network error*”, oleh karena itu data yang dimasukkan kedalam *fishbone diagram* adalah data yang berkaitan saja. Untuk jumlah keseluruhan data penyebab yang didapat yakni sebesar 1643 yang dibagi ke dalam tiga *consequent*. Kemudian untuk data sub penyebab yang dimasukkan ke *fishbone diagram* ialah data yang mana penjelasan tentang masalah yang terdapat diaplikasi mendekati sempurna. Berikut merupakan penjelasan alasan tersebut :

Kata “login” memiliki dua sub penyebab atau bisa disebut alasan dari kepala penyebab. Dua alasan itu adalah “setelah update login gagal” dan “muncul pesan terjadi kesalahan saat login”. Untuk sub penyebab “setelah update login gagal”, pengguna mengalami kegagalan saat ingin login setelah melakukan update yang mana sebelum diupdate pengguna bisa melakukan login. Pada sub penyebab “muncul pesan terjadi kesalahan saat login” pengguna mendapatkan pesan bahwa terjadi kesalahan saat login. Kedua sub penyebab bisa masuk kedalam kategori “*app crash*”

Kata “daftar” memiliki satu sub penyebab yaitu “pengguna tidak mendapatkan kode verifikasi untuk menyelesaikan pendaftaran”. Pengguna saat ingin menyelesaikan pendaftaran diharuskan untuk mengisi kode verifikasi yang didapat dari bpjs ketenagakerjaan, namun setelah ditunggu kode verifikasi tidak langsung didapatkan. Sub penyebab ini bisa dimasukkan kedalam kategori “*network errors*”

Kata “*email*” memiliki sub penyebab berupa “kode verifikasi via email tidak bisa” dan “tidak menerima kode verifikasi via email”. Sub penyebab ‘kode verifikasi via email tidak bisa’ terjadi sesuai dengan namanya yakni pengguna ingin melakukan kode verifikasi lewat email tetapi tidak bisa. Untuk sub penyebab “tidak menerima kode verifikasi via email “ diketahui para pengguna tidak mendapatkan kode verifikasi lewat email. Kedua sub penyebab bisa masuk kedalam kategori “*network errors*”.



BAB 6

KESIMPULAN

6.1 Kesimpulan

Pada kesimpulan yang bisa diambil didasarkan dari tujuan penelitian aplikasi BPJSTKU, diantaranya yakni :

1. Hasil yang diperoleh berupa beberapa kata kunci yang menjadi masalah utama pada aplikasi BPJSTKU. Kata kunci tersebut seperti “login”, “daftar”, “email”, dan “bpjstku”. Kata kunci tersebut dipilih berdasarkan jumlah kemunculan berulang dari ulasan pengguna aplikasi BPJSTKU. Kata dengan tingkat kemunculan berulang tinggi memiliki indikasi bahwa kata tersebut dianggap sebagai salah satu permasalahan yang ada pada aplikasi BPJSTKU, sehingga kata tersebut berubah menjadi kata kunci. Kata kunci dengan tingkat kemunculan yang tinggi dimulai dari kata “aplikasi”, “login”, “daftar”, “email”, dan “bpjstku”.
2. Kata kunci yang dihasilkan tidak seluruhnya digunakan. Beberapa akan dimasukkan kedalam diagram *fishbone* sesuai dengan kategori *key performance indicator of mobile application*. Kata kunci tersebut adalah “login”, “daftar” dan “email”. Kata kunci atau bisa disebut sebagai penyebab setelah masuk di *fishbone diagram* memiliki sub penyebab yang menjelaskan mengapa penyebab bisa terjadi. Sub penyebab untuk login berupa “setelah update login gagal” dan “muncul pesan terjadi kesalahan saat login” kedua sub penyebab tersebut masuk dalam kategori ‘*app crash*’ dari *key performance indicator of mobile application*. Untuk penyebab daftar memiliki satu sub penyebab yakni “pengguna tidak mendapatkan kode verifikasi untuk menyelesaikan pendaftaran”. Sub penyebab tersebut bisa masuk ke dalam kategori “*network error*” dari *key performance indicator of mobile application*. Kemudian yang terakhir, penyebab “email” memiliki dua sub penyebab yaitu “kode verifikasi via email tidak bisa” dan “tidak

menerima kode verifikasi via email “. Sub penyebab tersebut bisa masuk kedalam kategori “*network error*” dari *key performance indicator of mobile application*.

6.2 Saran

Saran atau usulan untuk penelitian ini berupa saran perbaikan pada aplikasi BPJSTKU dan saran untuk penelitian selanjutnya. Berikut saran tersebut :

1. Perbaikan untuk aplikasi BPJSTKU selain berdasarkan kata kunci yang didapat dari ulasan pengguna aplikasi BPJSTKU, penggunaan metode *association rules*, dan penggunaan *fishbone diagram*, diperlukan juga penyelesaian masalah secara langsung pada aplikasi BPJSTKU dengan menggunakan cara *trial & error* dibantu dengan para ahli di bidang keilmuan yang sesuai dengan permasalahan yang terjadi.
2. Untuk penelitian kedepannya yaitu berupa mengikutsertakan ulasan dengan peringkat 3 dengan menggunakan *text mining* berdasar *lexicon*. Tujuan penggunaan *text mining* berdasar *lexicon* yaitu dapat memillah ulasan dengan konotasi positif dan negatif, hal ini dilakukan karena ulasan dengan peringkat 3 berkemungkinan memuat ulasan dengan konotasi negatif.

DAFTAR PUSTAKA

- Agrawal, R. (1996). Fast Discovery of Association Rules.
- Aliguliyev, R. M. (2015). Detecting Terrorism-Related Articles On The E-government Using Text-Mining Techniques. *Problem of information technology*, 36-46.
- Athey, S. (2019). The Impact of Machine Learning on Economics. *The Economics of Artificial Intelligence: An Agenda*, 507-547.
- Bachani, N. (2020, April 11). *Chunking in NLP Decoded*. (online):
<https://towardsdatascience.com/chunking-in-nlp-decoded-b4a71b2b4e24>
 (diakses tanggal 21 oktober 2021)
- BPS. (2020, Desember 2). *publication*. (online):
<https://www.bps.go.id/publication/2020/12/02/be999725b7ae62d84c6660/statistik-telekomunikasi-indonesia-2019.html>
 (diakses tanggal 21 Oktober 2021)
- BPS. (2021, Januari 21). *Hasil Sensus Penduduk 2020*. (online):
<https://www.bps.go.id/pressrelease/2021/01/21/1854/hasil-sensus-penduduk-2020.html> (diakses tanggal 21 Oktober 2021)
- BUMN. (2021, juni 26). *lihat*. (online):
<https://jdih.bumn.go.id/lihat/UU%20Nomor%2024%20Tahun%202011>
 (diakses tanggal 21 Oktober 2021)
- Cermati. (2019, September). *Bpjs Ketenagakerjaan Program dan Cara Mendaftarnya*. (online): <https://www.cermati.com/artikel/bpjs-ketenagakerjaan-program-dan-cara-mendaftarnya>
 (diakses tanggal 21 Oktober 2021)
- Chakravarthy, S. (2020, Juni 19). *Tokenization for natural language processing*. (online): <https://towardsdatascience.com/tokenization-for-natural-language-processing-a179a891bad4>
 (diakses tanggal 21 Oktober 2021)
- Cheng, L. C. (2020). Analysing Digital Banking Reviews Using Text Mining. *2020 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining* (pp. 914-918). Hague: IEEE.
- Dzulfikar. (2019, Juni 7). *BPJS Ketenagakerjaan*. (online):
<https://lifepal.co.id/media/bpjs-ketenagakerjaan-mobile/>
 (diakses tanggal 21 Oktober 2021)

- Garg, A. (2018, september 3). *association rules*. (online):
<https://towardsdatascience.com/association-rules-2-aa9a77241654>
(diakses tanggal 21 Oktober 2021)
- Habib, A. (2018). Success Measures Evaluation for Mobile Commerce Using Text Mining based on Customer Tweets. *IOP Conf. Series: Materials Science and Engineering 319* (pp. 1-6). Yogyakarta: IOP.
- Hahsler, M. (2017). Visualizing association rules in hierarchical groups. *Journal of Business Economics*, 317-335.
- Hearst, M. (2003, Oktober 17). *~hearst/text-mining*. (online):
<https://people.ischool.berkeley.edu/~hearst/text-mining.html>
- Heidenreich, H. (2018, Desember 22). (online):
<https://towardsdatascience.com/stemming-lemmatization-what-ba782b7c0bd8>
(diakses tanggal 21 Oktober 2021)
- Hussein, N. (2015). Using the interestingness measure lift to generate association rules. *Journal of Advanced Computer Science & Technology*, 156-162.
- IBM, E. C. (2020, July). *Machine Learning*. (online):
<https://www.ibm.com/cloud/learn/machine-learning>
(diakses tanggal 21 Oktober 2021)
- Idris, M. (2021, Maret 10). *money.kompas*. (online):
<https://money.kompas.com/read/2021/03/10/092010926/cara-cek-saldo-bpjs-ketenagakerjaan-via-online-dan-sms?page=all>
(diakses tanggal 21 Oktober 2021)
- Islam, R. (2010). Mobile Application and Its Global Impact. *International Journal of Engineering & Technology*, 72-78.
- Jin, Y. (2017). Development of Word Cloud Generator Software Based on Python. *Procedia Engineering*, 788-792.
- KPI. (2022). *KPI-Basics*. (online): <https://kpi.org/KPI-Basics>
(diakses tanggal 4 Februari 2022)
- Lai, Y.-H. (2017). Analysis of User Feedback in The Mobile App Store Using Text Mining: A Case Study of Goole Fit. *2017 IEEE 8th International Conference on Awareness Science and Technology* (pp. 50-54). Taichung: IEEE.

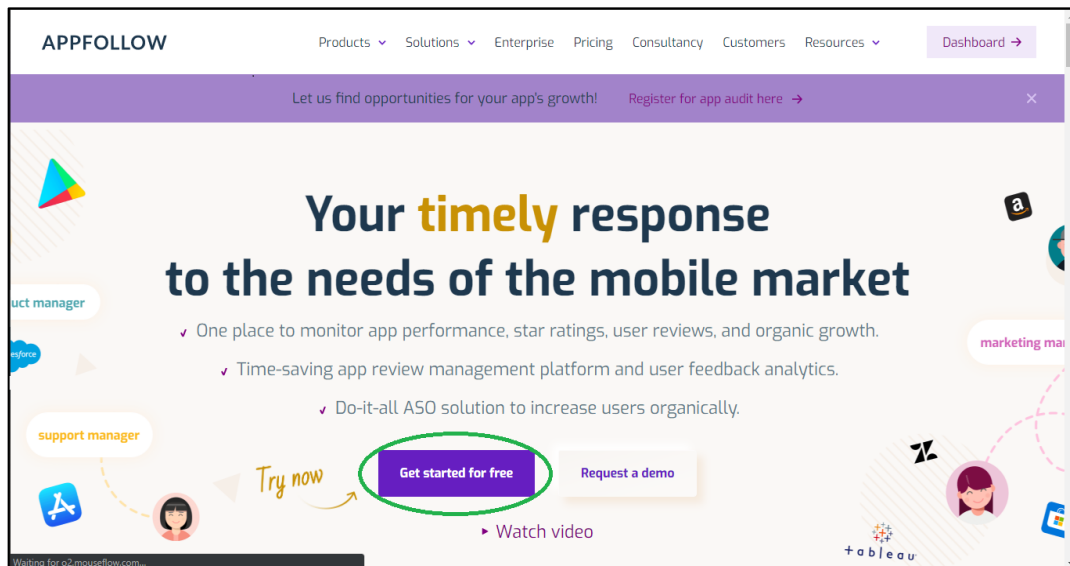
- Lekham, L. A. (2021). A Root Cause Analysis Using Text Mining for Patient No-Shows : A Case Study of Primary Care Centers Serving Rural Areas. *IISE Annual Conference*, (pp. 1-6).
- Liliana, L. (2016). A new model of ishikawa diagram for quality assessment. *IOP Conferenc Series* (hal. 1-7). IOP Publishing.
- Lumanauw, N. (2021, Februari 22). *anggoro-akan-digitalisasi-jaminan-sosial-bpjs-ketenagakerjaan*. (online):
<https://www.beritasatu.com/nasional/736823/anggoro-akan-digitalisasi-jaminan-sosial-bpjs-ketenagakerjaan>
 (diakses tanggal 21 Oktober 2021)
- Lutkevich, B. (n.d.). *definition: association-rules-in-data-mining*. (online):
<https://searchbusinessanalytics.techtarget.com/definition/association-rules-in-data-mining> (diakses tanggal 21 Oktober 2021)
- Mary, T. (2020, Februari 5). *Text Mining a Basic Beginners Guide*. (online):
<https://medium.com/budding-data-scientist/text-mining-a-basic-beginners-guide-5d2ecc04e4e6> (diakses tanggal 21 Oktober 2021)
- Metsker, O. (2018). Text and Data Mining Techniques in Judgment Open Data Analysis for Administrative Practice Control. *EGOSE*. St. Petersburg: Springer.
- Mirchandani, D. A. (2008). Perspectives of citizens towards e-government in thailand and indonesia: A multigroup analysis. *Information System Frontiers*, 483-497.
- Muhammad, I. (2021, 03 10). *cara-cek-saldo-bpjs-ketenagakerjaan-via-online-dan-sms*. (online): <https://money.kompas.com/read/2021/03/10/092010926/cara-cek-saldo-bpjs-ketenagakerjaan-via-online-dan-sms?page=all> (diakses tanggal 21 Oktober 2021)
- Najafabadi, M. K. (2017). Improving the accuracy of collaborative filtering recommendations using clusterng and association rules mining on implicit data. *Computer in Human Behavior*, 113-128.
- Nisa, C. (2020). Evaluasi Implementasi Aplikasi Bpjsku Mobile. *Buletin Penelitian Kesehatan*, 51-62.
- Nuraini, T. N. (2021, April 27). *fungsi-bpjs-ketenagakerjaan-dapat-tingkatkan-produktivitas-para-pekerja*. (online): <https://www.merdeka.com/trending/fungsi-bpjs-ketenagakerjaan-dapat-tingkatkan-produktivitas-para-pekerja.html?page=all> (diakses tanggal 21 Oktober 2021)

- Nurhayati-Wolff, H. (2020, Juli). *Mobile Internet & Apps*.(online):
<https://www.statista.com/statistics/309017/indonesia-mobile-phone-internet-user-penetration/> (diakses tanggal 21 Oktober 2021)
- Nurhayati-Wolff, H. (2021, Agustus 24). *market-share-held-by-mobile-operating-systems-in-indonesia*. (online):
<https://www.statista.com/statistics/262205/market-share-held-by-mobile-operating-systems-in-indonesia/> (diakses tanggal 21 Oktober 2021)
- Nurrahmi, S. (2018, september 4). *bpjs-ketenagakerjaan-digitalisasi-vs-efisiensi*. (online): <https://www.finansialku.com/bpjs-ketenagakerjaan-digitalisasi-vs-efisiensi/> (diakses tanggal 21 Oktober 2021)
- Payton, F. C. (2018). Text Mining Mental Health Health Reports for Issues Impacting Today's College Students: Qualitative Study. *JMIR MENTAL HEALTH*, 1-12.
- Putri, A. S. (2020, Januari). *Edukasi Skola*. (online):
<https://www.kompas.com/skola/read/2020/01/15/160000469/bpjs-ketenagakerjaan--sejarah-singkat-jenis-visi-dan-misi?page=all> (diakses tanggal 21 Oktober 2021)
- Pykes, K. (2020, November 26). *part of Speesch tagging fo Beginners*. (online):
<https://towardsdatascience.com/part-of-speech-tagging-for-beginners-3a0754b2ebba> (diakses tanggal 21 Oktober 2021)
- Rakhmayanti, I. (2020, November). *penggunaan-smartphone-di-indonesia-meningkat-49-di-q3-2020-1606482674*. (online):
<https://tekno.sindonews.com/read/248398/122/penggunaan-smartphone-di-indonesia-meningkat-49-di-q3-2020-1606482674> (diakses tanggal 21 Oktober 2021)
- Riyanto, G. P. (2021, Februari 23). *jumlah-pengguna-internet-indonesia-2021-tembus-202-juta*. (online): <https://tekno.kompas.com/read/2021/02/23/16100057/jumlah-pengguna-internet-indonesia-2021-tembus-202-juta> (diakses tanggal 21 Oktober 2021)
- Sabani, A. D. (2018). A conceptual framework for the adoption of e-government in Indonesia. *Australasian Conference on Information Systems*. Sydney.
- Sinha, A. A. (2020). Improving the service quality of telecommunication companies using online customer and employe review analysis. *Quality Management Journal*, 182-199.

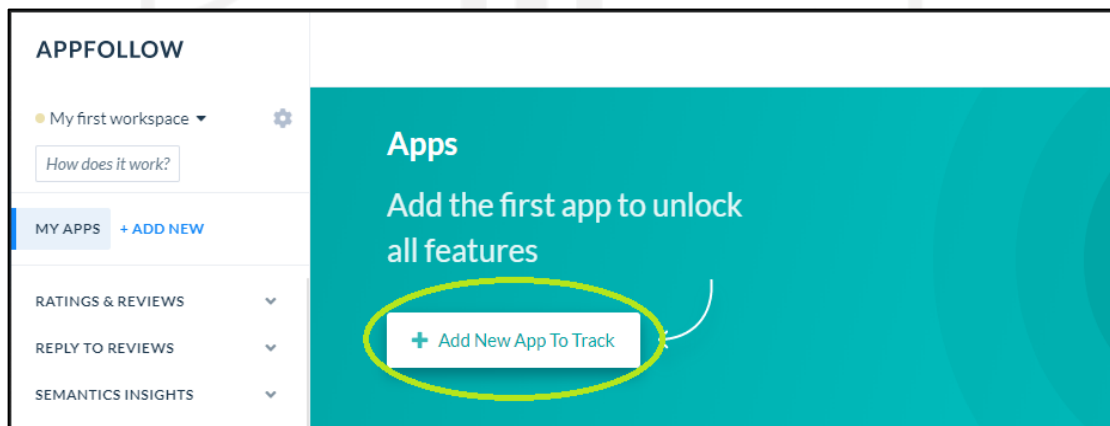
- Smith, A. (2019, October 18). *amp*. (online): <https://yourstory.com/mystory/mobile-app-performance-metrics-to-gauge-app-success/amp>
(diakses tanggal 4 Februari 2022)
- Tempo.co. (2021, Maret 13). *sebelum-klaim-bpjs-ketenagakerjaan-pelajari-syarat-dan-tahapannya*. (online): <https://bisnis.tempo.co/read/1441741/sebelum-klaim-bpjs-ketenagakerjaan-pelajari-syarat-dan-tahapannya/full&view=ok>
(diakses tanggal 21 Oktober 2021)
- Wijaya, A. P. (2016). Naive Bayes Classification pada Klasifikasi Dokumen Untuk Identifikasi Konten E-Government. *Journal of Applied Intelligent System*, 48-55.
- Zhou, H. (2019). User-opinion mining for mobile library apps in China: exploring user improvement needs. *Library Hi Tech*, 325-337.
- Zhou, V. (2019, Desember 11). *A Simple Explanation of The Bag Of Words Model*. (online): <https://towardsdatascience.com/a-simple-explanation-of-the-bag-of-words-model-b88fc4f4971>
(diakses tanggal 21 Oktober 2021)

LAMPIRAN

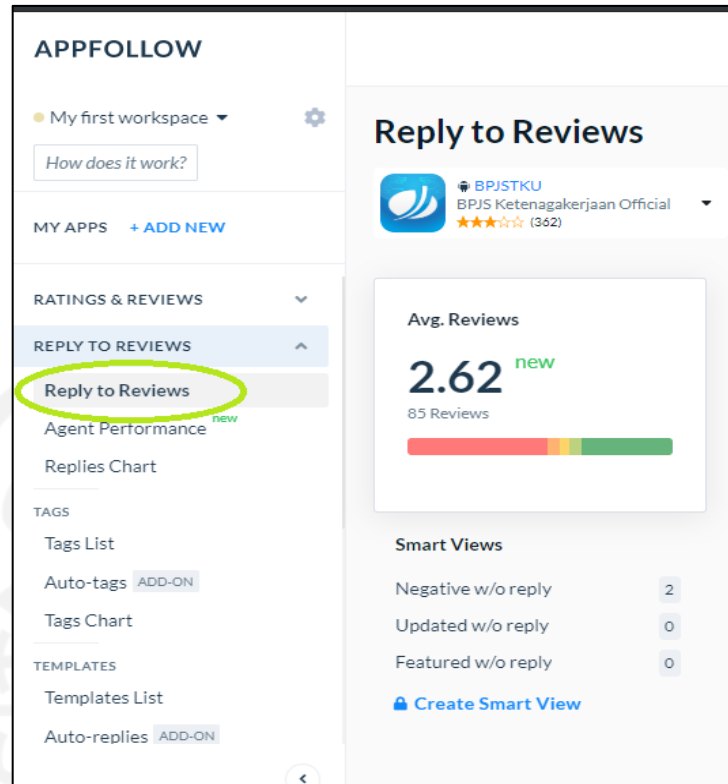
Lampiran 1. Pengumpulan data mentah menggunakan teknik *web scrapping* dibantu dengan Aplikasi AppFollow



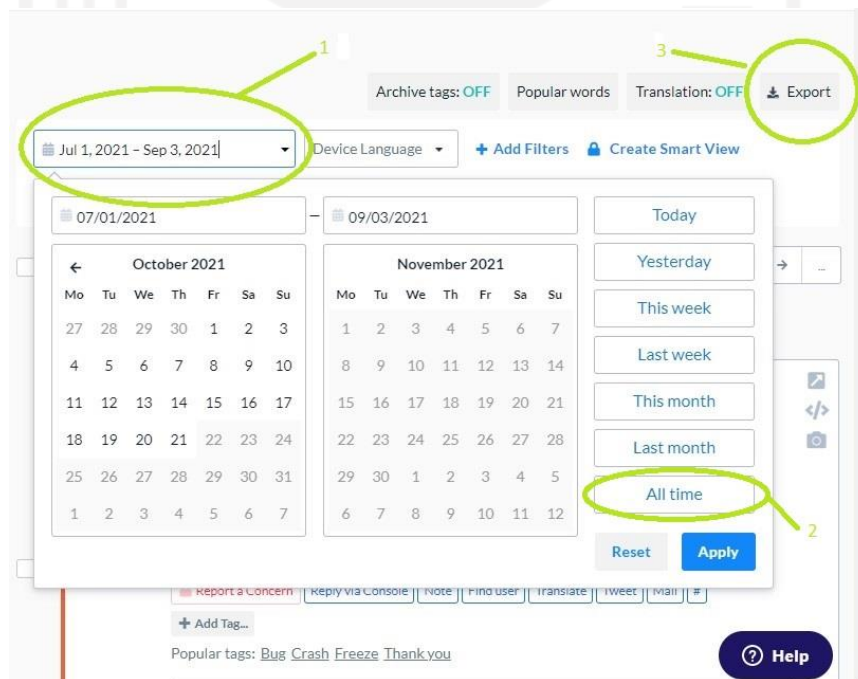
Gambar L. 1 Halaman Utama AppFollow



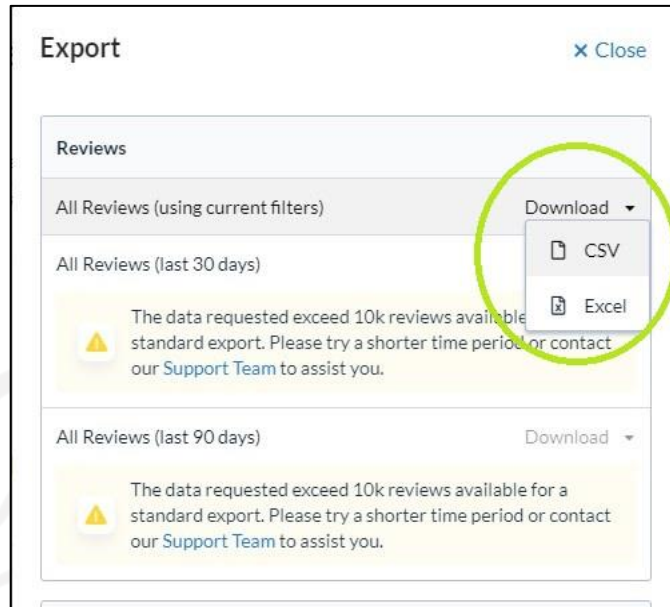
Gambar L. 2 Memilih Aplikasi



Gambar L. 3 Menuju Reviews Para Pengguna BPJSTKU



Gambar L. 4 Memilih Tanggal Untuk Data Reviews yang kemudian di Export



Gambar L. 5 Download All Reviews dan Pilih CSV



Lampiran 2. Source Code Proses Pengolahan Data

Import Library

```
#Pandas dan Numpy
```

```
import pandas as pd
```

```
import numpy as np
```

```
#Nltk
```

```
import nltk
```

```
from nltk.corpus import stopwords
```

```
stopwords_indonesia = set(stopwords.words('indonesian'))
```

```
#Matplotlib
```

```
import matplotlib.pyplot as plt
```

```
import matplotlib.image as mpimg
```

```
from matplotlib.ticker import PercentFormatter
```

```
%matplotlib inline
```

```
#Wordcloud
```

```
from wordcloud import WordCloud, ImageColorGenerator
```

```
from PIL import Image
```

```
#Association Rule
```

```
from mlxtend.frequent_patterns import association_rules, apriori
```

```
from mlxtend.preprocessing import TransactionEncoder
```

Text Preprocessing / Memasukkan data kedalam Jupyter Notebook

```
#Pemanggilan file csv
```

```
df=pd.read_csv('GG.csv', sep=";", header=0, usecols=['Review', 'Rating'])
df
```

```
#Penggantian kata rating dan review di seluruh kolom menjadi NaN value
```

```
df1=df.apply(lambda x: x.str.replace( r'\W', ' ').str.lower()).replace(('rating', 'review'),
np.nan, regex=True)
df1
```

```
#Penghapusan baris yang berisikan NaN value
```

```
df1= df1.dropna().reset_index(drop=True)
```

```
#Pembuatan kolom baru bernama 'bad_review' yang hanya berisi rating bintang 1 dan 2
```

```
df1['bad label']=df1['Rating']
```

```
df1['Rating']=df1['Rating'].astype('int8')
```

```
df1['bad label']=df1['bad label'].astype('int8')
```

```
df1['bad label']=df1['Rating'].apply(lambda x:x if(x < 3) else None)
```

```
#Membuang data selain rating 1, dan 2 serta mengganti ukuran file data menjadi ke
terendah
```

```
df2=df1.dropna().reset_index(drop=True)
```

```
df3=df2
```

```
df3['bad label']=df3['bad label'].astype('int8')
```

Tokenization sebelum stopwords dan perbaikan dan penggabungan kata

#Perubahan dari dataframe menjadi kumpulan kata agar bisa diproses pada *wordcloud*

```
token=' '.join(df2['Review'])
```

#Mencari kata tak berarti, rusak, dan memiliki makna yang sama dengan *wordcloud*

```
wc = WordCloud(background_color = 'white',
               width = 800, height = 800,
               min_font_size = 10,
               font_path = 'arial',
               collocations = False,
               max_words = 500).generate(token)
```

```
plt.figure(figsize = (10,10))
```

```
plt.imshow(wc)
```

```
plt.axis('off')
```

```
plt.show()
```

#Menggunakan *Barchart* untuk mengetahui lebih detail kata mana yang akan dibuang dan di perbaiki dan digabung

```
n = 16
```

```
dummy = nltk.tokenize.word_tokenize(token)
```

```
dummy_Word=nltk.FreqDist(dummy)
```

```
dummy_Word2=nltk.FreqDist(w for w in dummy)
```

```
#tabel
```

```
print('          BPJSTKU          ')
print('='*60)
```

```
hasil1=pd.DataFrame(dummy_Word2.most_common(n), columns=['Kata', 'Frekuensi'])
print(hasil1)
print('='*60)
```

```
hasil1=pd.DataFrame(dummy_Word2.most_common(n), columns=['Kata',
'Frekuensi']).set_index('Kata')
```

```
hasil1.plot.bar(rot=0, figsize=(12, 6), width=0.5)
```

Wordcloud dengan hasil data yang diinginkan

```
#stopword
```

```
SW = ['dan', 'dan saya', 'saya', 'yg', 'tapi', 'nya', 'ini', 'di', 'untuk', 'bikin', 'kali', 'trus', 'udh',
'kesalahan', 'tolong', 'yang', 'setelah', 'dibuka', 'malah', 'ada', 'lagi', 'kartu', 'ni', 'fitur', 'isi',
'dgn', 'bgt', 'clear', 'cache', 'solusi', 'solusinya', 'berkali', 'langsung', 'banget', 'kalo', 'kaya',
'dicoba', ',', '1.0', '2.0', '!', ';', '1', 'i', 't', '2', '3', '5', '4', 'gimana', 'pas', 'sih', 'hp', 'terima', 'e',
'd', 'g', 'open', 'uang', 'digital', 'my', 'found', '...', 'saldo', 'coba', 'no', 'ko', 'udh', 'antrian', 'ga',
'a', 's', 'b', 'membantu', 'sampah', 'when', 'bi', 'ya', 'tp', 'it', 'apk', 'mulu', 'online', 'kasih',
'bintang', 'pop', 'up', 'is', '()', 'aja', 'mau', 'buka', 'cek', 'masuk', 'data', 'mohon', 'gak', 'gk',
'tdk', 'ganti', 'giliran', 'tidak', 'udah', 'bisa', 'blm', 'bs', 'klo', 'lg', 'sdh', 'sy', 'bsa', 'knp', 'min',
'nih', 'sy', 'gini', 'mending', 'gini', 'bsa', 'nih', 'suruh', 'versi', 'tulisan', 'muncul', 'min', 'ulang',
'tetep', 'orang', 'masukin', 'ngecek']+ list(stopwords_indonesia)
```

```
#Perbaiki dan penggabungan kata
```

```
kata_perbaikan = {'force':'forceclose', 'close':'forceclose', 'forced':'forceclose',
'closed':'forceclose', 'eror':'error', 'log':'login', 'in':'login', 'login':'login', 'errorrrrrrr':'error',
'error':'error', 'bpjs':'bpjstku', 'app':'aplikasi', 'aplikasinya':'aplikasi', 'apps':'aplikasi',
'application':'aplikasi', 'bpjstku':'bpjstku', 'tdk':'tidak', 'udah':'sudah', 'aplikasi':'aplikasi',
'aplikasi':'aplikasi', 'instal':'install', 'install':'install', 'ketenagakerjaan':'bpjstku',
'pake':'pakai', 'uninstal':'uninstall', 'terdaftar':'daftar'}
```

```
code = {r'(\b){}(\b)'.format(call):r'\1{ }2'.format(wind) for call,wind in
kata_perbaikan.items()}
```

```
df3['Review'] = df3['Review'].replace(code, regex=True)
```

```
#Tokenization ulang
```

```
token2=' '.join(df3['Review'])
```

```
#kata-kata yang diinginkan mulai kelihatan di wordcloud
```

```
wc = WordCloud(background_color = 'white',
width = 800, height = 800,
min_font_size = 10,
font_path = 'arial',
collocations = False,
max_words = 500,
stopwords = set(SW)).generate(token2)
```

```
plt.figure(figsize = (10,10))
```

```
plt.imshow(wc)
```

```
plt.axis('off')
```

```
plt.show()
```

#Pembuatan *bar chart* untuk mengetahui jumlah kata yang diinginkan di *wordcloud* secara detail

```

n = 5
Kata_token = nltk.tokenize.word_tokenize(token2)
Wordf=nltk.FreqDist(Kata_token)
Word_except_stop_dist=nltk.FreqDist(w for w in Kata_token if w not in SW)

#tabel

print('          BPJSTKU          ')
print('='*60)

hasil2=pd.DataFrame(Word_except_stop_dist.most_common(n), columns=
                    ['Kata', 'Frekuensi'])
print(hasil2)
print('='*60)

hasil2=pd.DataFrame(Word_except_stop_dist.most_common(n),          columns=['Kata',
'Frekuensi']).set_index('Kata')

hasil2.plot.bar(rot=0, figsize=(12, 6), width=0.5)

plt.xlabel("Kata")
plt.ylabel("Frekuensi")
plt.title("")
plt.xticks(fontsize=15)
plt.show()

```

Data Mining dengan menggunakan teknik Association Rules

#koding untuk bisa masuk ke koding association rules. Data sudah dibersihkan dari kata yang tidak berarti dan penggabungan kata sudah dilakukan

```
df3['Review'] = df3['Review'].apply(lambda x: ' '.join([word for word in x.split() if word not in (SW)]))
```

```
#Menghilangkan semua kolom kecuali kolom ulasan
```

```
df4=df3.drop(labels=['Rating', 'bad label'], axis=1)
```

```
#Pembuatan multidimensional list
```

```
dataset=[w for w in df4['Review'].str.split(' ')]
```

```
dataset
```

```
#Pelabelan true atau false. true = jika satu kata sama antar baris dan kolom, false = sebaliknya true
```

```
te = TransactionEncoder()
```

```
te_ary = te.fit_transform(dataset)
```

```
df5 = pd.DataFrame(te_ary, columns=te.columns_)
```

```
#Mengetahui kata yang sering ada pada dataset di koding association rules
```

```
ap = apriori(df5, min_support= 0.005, use_colnames=True)
```

```
#Mengetahui apakah ada hubungan antara satu kata dengan kata lain
```



```

ar = association_rules(ap, metric='lift', min_threshold=1)
arr = pd.DataFrame(ar)
arr

#Menghilangkan kolom yang tidak digunakan

ar1=arr.drop(labels=['antecedent support', 'consequent support', 'leverage',
'conviction'], axis=1)

#Membuka data hubungan antar kata

with pd.option_context('display.max_rows', 9000):
    print(ar1)

##### Memasukkan ke Fishbone diagram#####

#Pandas dan Numpy

import pandas as pd
import numpy as np

df=pd.read_csv('GG.csv', sep=";", header=0, usecols=['Review', 'Rating'])

df1=df.apply(lambda x: x.str.replace( r'\W', ' ').str.lower()).replace(('rating', 'review'),
np.nan, regex=True)

df1= df1.dropna().reset_index(drop=True)

df1['bad label']=df1['Rating']

df1['Rating']=df1['Rating'].astype('int8')

```

```
df1['bad label']=df1['bad label'].astype('int8')
```

```
df1['bad label']=df1['Rating'].apply(lambda x:x if(x < 3) else None)
```

```
df2=df1.dropna().reset_index(drop=True).drop(labels=['Rating', 'bad label'], axis=1)
```

```
df22 = df2.dropna()
```

```
df3=df2
```

```
df22[df22.Review.str.contains('lupa')]
```

```
df22[~df22.Review.str.contains('lupa')]
```

```
df22.dropna()
```

```
df21 = df22.dropna()
```

```
df31 = df21
```

```
df31=df21['Review'].apply(lambda x: ''.join([word for word in x.split()]))
```

```
df41 = pd.DataFrame(df31)
```

```
df41
```

```
def containslidf41(x):
```

```
    i = df41[df41['Review'].str.contains(x)]
```

```
    return i
```

kp = {'nik' : 'ktp'
 , 'nmr': 'nomor'
 , 'memaukkan': 'memasukkan'
 , 'niknya' : 'ktp'
 , 'g' : 'tidak'
 , 'gk' : 'tidak'
 , 'ga' : 'tidak'
 , 'gx' : 'tidak'
 , 'gak' : 'tidak'
 , 'nggk' : 'tidak'
 , 'nga' : 'tidak'
 , 'nggak' : 'tidak'
 , 'tdk' : 'tidak'
 , 'bs' : 'bisa'
 , 'bsa' : 'bisa'
 , 'bis' : 'bisa'
 , 'full' : 'penuh'
 , 'apk' : 'aplikasi'
 , 'appnya' : 'aplikasi'
 , 'appl' : 'aplikasi'
 , 'app' : 'aplikasi'
 , 'aplikasinya' : 'aplikasi'
 , 'apps' : 'aplikasi'
 , 'application' : 'aplikasi'
 , 'bpjstku' : 'aplikasi'
 , 'gacor' : ''
 , 'muter' : 'loading'
 , 'muter2' : 'loading'
 , 'pdhal' : 'padahal'
 , 'pd hal' : 'padahal'
 , 'yg' : 'yang'
 , 'mending yang lama' : 'lebih_baik_aplikasi_versi_dulu'

```

, 'pake yang lama'      : 'lebih_baik_aplikasi_versi_dulu'
, 'bagusan'            : 'bagusan'
, 'bagusan yang lama'  : 'lebih_baik_aplikasi_versi_dulu'
, 'di aplikasi lama'   : 'di_aplikasi_versi_dulu'
, 'aplikasi versi lama' : 'aplikasi_versi_dulu'
, 'perusahaan lama'    : 'perusahaan_dulu'
, 'semakin lama'       : 'semakin_tambah_umur'
, 'klaim lama'         : 'klaim_lamban'
, 'register'           : 'daftar'
, 'tidak tersedia'     : 'tidak_tesedia'
, 'padahal sinyal'     : 'padahal_sinyal'
, 'bagusan'            : 'lebih_baik'
, 'sinyalnya'          : 'sinyal'
, 'sinyall'            : 'sinyal'
, 'ane'                : 'saya'
, 'sy'                 : 'saya'
, 'sya'                : 'saya'
, 'padahal sinyal'     : 'padahal_sinyal'
, 'loding'             : 'loading'
, 'lodingnya'          : 'loading'
, 'lodingnye'          : 'loading'
, 'lodingny'           : 'loading'
, 'lodingya'           : 'loading'
}

```

```
co1 = {r'(\b){(\b)'.format(call1):r'\1{\}\2'.format(wind1) for call1,wind1 in kp.items()}
```

```
df411 = df41['Review']=df41['Review'].replace(co1, regex=True)
```

```

kp2 = { 'loading terus'      : 'loading_nya_lama'
, 'loading teruss'          : 'loading_nya_lama'
, 'loading terusss'         : 'loading_nya_lama'

```

```

, 'loading terussss' : 'loading_nya_lama'
, 'loading trs' : 'loading_nya_lama'
, 'log in' : 'login'
, 'log ini' : 'login'
, 'log in2' : 'login'
, 'log innya' : 'login'
, 'log innj' : 'login'
, 'login keluar sendiri' : 'login forceclose'
, 'aplikasi yang lama' : 'aplikasi_versi_dulu'
, 'aplikasi yang lamanya' : 'aplikasi_versi_dulu'
, 'mending' : 'lebih_baik'
, 'bukanya' : 'bukan'
, 'mendingan' : 'lebih_baik'
, 'bagus2' : 'bagus'
, 'bagus2' : 'bagus'
, 'udah bagus' : 'sudah_lebih_baik'
, 'di bagusin' : 'dibagusin'
, 'dibagusin' : 'diperbaiki'
, 'bukannya' : 'bukan'
, 'bukan nya' : 'bukan'
, 'bukan semakin bagus' : 'bukan_menjadi_lebih_baik'
}

```

```
co2 = {r'(\b){}(\b)'.format(call2):r'\1{}'\2'.format(wind2) for call2,wind2 in kp2.items()}
```

```
df411 = df41['Review']=df41['Review'].replace(co2, regex=True)
```

```

kp3 = {'hebat sekali ini aplikasi saking hebatnya' : 'aplikasi_tidak_bisa'
, 'saking' : 'saking'
, 'ny' : 'nya'
, 'nya' : ''
, 'loadingnya' : 'loading_nya'

```

```

, 'loadingnyaaaaa'          : 'loading_nya'
, 'loadingnyaa'            : 'loading_nya'
, 'kok'                    : ''
, 'eror'                   : 'error'
, 'erorr'                  : 'error'
, 'errorrrr'              : 'error'
, 'network'                : 'jaringan'
, 'networking'            : 'jaringan'
, 'notnetwork'            : 'tidak_ada_jaringan'
, 'jaringan penuh'        : 'sinyal_lancar'
, 'jaringan 4g'          : 'sinyal_lancar'
, 'jaringan bagus'       : 'sinyal_lancar'
, 'jaringan error'       : 'muncul_notifikasi_jaringan_error'
, 'muncul notif'         : 'muncul_notifikasi'
, 'muncul notifikasi'    : 'muncul_notifikasi'
, 'terhenti'             : 'berhenti'
, 'error 404'            : 'error_404'
, 'forced'               : 'force'
, 'closed'              : 'close'
, 'force close'         : 'force_close'
, 'forceclose'         : 'force_close'
, 'forceclosed'        : 'force_close'
, 'forcedclose'        : 'force_close'
, 'forcedclosed'       : 'force_close'
, 'hp yang lama'       : 'hp_yang_dulu'
, 'dihp yang lama'    : 'di_hp_yang_dulu'

}

```

```
co3 = {r'(\b){ }(\b)'.format(call3):r'\1 { } \2'.format(wind3) for call3,wind3 in kp3.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co3, regex=True)
```

kp4 = {'bergunanya':'berguna'
 , 'bergunaaaa':'berguna'
 , 'no':'nomor'
 , 'lamaa':'lama'
 , 'lamaaa':'lama'
 , 'lamaaaa':'lama'
 , 'lamaaaaa':'lama'
 , 'lamaaaaaa':'lama'
 , 'lamaaaaaaa':'lama'
 , 'lamaaaaaaaa':'lama'
 , 'lamaaaaaaaaa':'lama'
 , 'lamaaaaaaaaaa':'lama'
 , 'lamaaaaaaaaaaa':'lama'
 , 'yg':'yang'
 , 'yng':'yang'
 , 'loudingnya':'loading'
 , 'email yang lama':'email_yang_dulu'
 , 'diemail yang lama':'email_yang_dulu'
 , 'akun yang lama':'akun_yang_dulu'
 , 'bpjstk yang lama':'aplikasi_yang_versi_dulu'
 , 'klamaan':'terlalu_lama'
 , 'lamaan':'terlalu_lama'
 , 'kelamaan':'terlalu_lama'
 , 'data yang lama':'data_yang_dulu'
 , 'kerja yang lama':'tempat_kerja_yang_dulu'
 , 'sandi yang lama':'sandi_yang_dulu'
 , 'pt lama':'tempat_kerja_yang_dulu'
 , 'nol besar':'tidak_memuaskan'
 , 'bintang nol':'bintang_nol'
 , 'hasil nol':'hasil_nol'
 , 'sudah nol':'sudah_nol'
 , 'tidak berguna':'tidak_berguna'
 , 'pt yang lama':'tempat_kerja_yang_dulu'

```
}
```

```
co4 = {r'(\b){ }(\b)'.format(call4):r'\1{ }2'.format(wind4) for call4,wind4 in kp4.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co4, regex=True)
```

```
kp5 = { 'root device':'root_device'
      , 'device':'android'
      , 'bpjs yang lama':'aplikasi_versi_dulu'
      , 'versi yang lama':'aplikasi_versi_dulu'
      , 'aplikasi yang pertama':'aplikasi_versi_dulu'
      , 'ktp yang lama':'ktp_yang_dulu'
      , 'kpj yang lama':'kpj_yang_dulu'
      , 'perusahaan yang lama':'tempat_kerja_yang_dulu'
      , 'diperusahaan yang lama':'tempat_kerja_yang_dulu'
      , 'gmail yang lama':'gmail_yang_dulu'
      , 'login email':'login_email'
      , 'terdaptar':'terdaftar'
      , 'terdaftarlah':'terdaftar'
      , 'terdaftaranya':'terdaftar'
      , 'terdaftart':'terdaftar'
      , 'terdaftartar':'terdaftar'
      , 'tidak terdaftar':'tidak_terdaftar'
      , 'nomee':'nomor'
      , "nomornya":'nomor'
      , 'nomer':'nomor'
      , 'diktpnya':'di ktp'
      , 'dikpj':'di kpj'
      , '1ktp':'ktp'
      , 'diktp':'di ktp'
      , 'iktp':'ktp' }
```



```

, 'aktp': 'ktp'
, 'ktpnya': 'ktp'
, 'ktpsaya': 'ktp'
, '1kpj': 'kpj'
, '3kpj': 'kpj'
, 'kjp': 'kpj'
, 'kjpnya': 'kpj'
, 'kpjnya': 'kpj'
, 'kpjku': 'kpj'
, 'ektp': 'e_ktp'
, 'e ktp': 'e_ktp'
}

```

```
co5 = {r'(\b){ }(\b)'.format(call5):r'\1{ } \2'.format(wind5) for call5, wind5 in kp5.items() }
```

```
df411 = df41['Review'] = df41['Review'].replace(co5, regex=True)
```

```

kp6 = { 'kpj': 'nomor kpj'
, 'ktp': 'nomor ktp'
, 'ktpsaja': 'nomor ktp'
, 'e_ktp': 'nomor_e_k_t_p'
, 'tgl': 'tanggal'
, 'tglnya': 'tanggal'
, 'lhr': 'lahir'
, 'tanggal lahir': 'tanggal_lahir'
, 'tgllahir': 'tanggal_lahir'
, 'tglblnthn': 'tanggal_lahir'
}

```

```
co6 = {r'(\b){ }(\b)'.format(call6):r'\1{ } \2'.format(wind6) for call6, wind6 in kp6.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co6, regex=True)
```

```
kp7 = {'nomor ktp':'nomor_k_t_p'
      , 'nomor kpj':'nomor_k_p_j'
      , 'bpjsnya':'bpjs'
      , 'bpjsny':'bpjs'
      , 'bpjsk':'bpjs'
      , 'bpjstk':'bpjs'
      , '2bpjs':'dua bpjs'
      , 'nomor kartu bpjs':'nomor_kartu_bpjs'
      , 'kartu bpjs':'nomor_kartu_bpjs'
      , '2kartu bpjs':'dua nomor_kartu_bpjs'
      , '3kartu bpjs':'tiga nomor_kartu_bpjs'
      , '4kartu bpjs':'empat nomor_kartu_bpjs'
      }
}
```

```
co7 = {r'(\b){ }(\b)'.format(call7):r'\1{ }2'.format(wind7) for call7, wind7 in
kp7.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co7, regex=True)
```

```
kp8 = {'kartu bpjs':'nomor_kartu_bpjs'
      , 'dikartu bpjs':'di nomor_kartu_bpjs'
      , '2kartu bpjs':'dua nomor_kartu_bpjs'
      , '3kartu bpjs':'tiga nomor_kartu_bpjs'
      , '4kartu bpjs':'empat nomor_kartu_bpjs'
      , 'kartu bpjsketenaga kerjaan': 'nomor_kartu_bpjs'
      , 'kartu bpjstenagakerjaan':'nomor_kartu_bpjs'
      , 'dikartu bpjs ketenagakerjaannya':'di nomor_kartu_bpjs'
      , 'kartu bpjsketenagaan kerja':'nomor_kartu_bpjs'
      }
```

```
}
```

```
co8 = {r'(\b){}\b'.format(call8):r'\1{}\2'.format(wind8) for call8, wind8 in
kp8.items()}
```

```
df411 = df41['Review']=df41['Review'].replace(co8, regex=True)
```

```
kp9 = {'bisa2':'bisa'
, 'bisa2':'bisa'
, 'bisaa':'bisa'
, 'begitu2':'begitu'
, 'jelas2':'jelas'
, 'jgn2':'jangan_jangan'
, 'jelas':'jelas'
, 'jelasz':'jelas'
, 'jelasss':'jelas'
, 'jelassss':'jelas'
, 'jelasssss':'jelas'
, 'jelassssss':'jelas'
, 'jelasssssss':'jelas'
, 'jelassssssss':'jelas'
, 'jelasssssssss':'jelas'
, 'jelassssssssss':'jelas'
, 'jelasssssssssss':'jelas'
, 'jelassssssssssss':'jelas'
, 'jelasssssssssssss':'jelas'
, 'jelassssssssssssss':'jelas'
, 'jelasssssssssssssss':'jelas'
, 'jelassssssssssssssss':'jelas'
, 'jelasssssssssssssssss':'jelas'
, 'bisanya':'bisa'
, 'bisaaa':'bisa'
, 'bisaaaa':'bisa'
, 'bisaaaaa':'bisa'
, 'bisaaaaaa':'bisa'
, 'bisaaaaaaa':'bisa'}
```

```

, 'bisaaaaaaaa': 'bisa'
, 'bisaaaaaaaa': 'bisa'
, 'bisaaaaaaaa': 'bisa'
, 'bisad': 'bisa'
, 'aplikasibtidak': 'aplikasi tidak'
, 'yatidak': 'ya tidak'
, 'bukatidak': 'buka tidak'
, 'tapitidak': 'tapi tidak'
, 'bisamengecek': 'bisa mengecek'
, 'ditidak': 'di tidak'
, 'sayatidak': 'saya tidak'
, 'bisavbuka': 'bisa buka'
, 'bisaya': 'bisa ya'
, 'bisan': 'bisa'
, 'bisah': 'bisa'
, 'kenapantidak': 'tidak'
, 'bisak': 'bisa'
, 'bisa2d': 'bisa di'
, 'tetapntidak': 'tetap tidak'
, 'bisayh': 'bisa ya'
, 'bisahh': 'bisa'
, 'jugatidak': 'juga tidak'
, 'bisamasuk': 'bisa masuk'
, 'bisadi': 'bisa di'
, 'bisa_bisa': 'bisa'
}

```

```

co9 = {r'(\b){ }(\b)'.format(call9):r'\1 { } \2'.format(wind9) for call9, wind9 in
kp9.items()}

```

```

df411 = df41['Review'] = df41['Review'].replace(co9, regex=True)

```

```
kp10 = {'tidak bisa':'tidak_bisa'
        , 'tidak jelas':'tidak_jelas'
        }
```

```
co10 = {r'(\b){}(\b)'.format(call10):r'\1{}\2'.format(wind10) for call10, wind10 in
        kp10.items()}
```

```
df411 = df41['Review']=df41['Review'].replace(co10, regex=True)
```

```
kp11 = {'password':'sandi'
        , 'passwordnya':'sandi'
        , 'passwod':'sandi'
        , 'passwr':'sandi'
        , 'paswr':'sandi'
        , 'psword':'sandi'
        , 'pssword':'sandi'
        , 'pswd':'sandi'
        , 'pswr':'sandi'
        , 'psword':'sandi'
        , 'psswd':'sandi'
        , 'pass':'sandi'
        , 'passnya':'sandi'
        , 'passwor':'sandi'
        , 'passord':'sandi'
        , 'passwodnya':'sandi'
        , 'passwaordnya':'sandi'
        , 'passwordnya':'sandi'
        , 'passaword':'sandi'
        , 'passwordlah':'sandi'
        , 'passwordnya':'sandi'
        , 'passowrd':'sandi'
        , 'passwird':'sandi'
        , 'password':'sandi'}
```

```
, 'password ':'sandi'
, 'passworld':'sandi'
, 'fasword':'sandi'
}
```

```
co11 = {r'(\b){}(\b)'.format(call11):r'\1{}\2'.format(wind11) for call11, wind11 in
kp11.items()}
```

```
df411 = df41['Review']=df41['Review'].replace(co11, regex=True)
```

```
kp12 = {'imail':'email'
, 'gmail':'email'
, 'gimail':'email'
, 'sandy':'sandi'
, 'sandinya':'sandi'
, 'dan pin':'dan_pin'
, 'dan sandi':'dan_sandi'
, 'udh':'sudah'
, 'sudh':'sudah'
, 'sudha':'sudah'
, 'dipermudah':'di permudah'
, 'udah':'sudah'
, 'udhh':'sudah'
, 'udha':'sudah'
, 'ssh':'susah'
, 'sush':'susah'
, 'ssah':'susah'
, 'sussahh':'susah'
, 'sussah':'susah'
, 'sussah':'susah'
, 'susah sekali':'susah'
, 'sajasaldo':'saja'
, 'pas mau login':'pas mau login'}
```

}

```
co12 = {r'(\b){}(\b)'.format(call12):r'\1{}\2'.format(wind12) for call12, wind12 in
kp12.items()}
```

```
df411 = df41['Review']=df41['Review'].replace(co12, regex=True)
```

```
kp13 = {'aja' :''
, 'amat' :''
, 'ajaa' :''
, 'ajaaa' :''
, 'ajaaaa' :''
, 'ajaaaaa' :''
, 'aaja' :''
, 'ya' :''
, 'saja' :''
, 'sajaa' :''
, 'sajaaa' :''
, 'sajaaaa' :''
, 'asaja' :''
, 'sajalah' :''
, 'sajak' :''
, 'sajah' :''
, 'saya' :''
, 'ko' :''
, 'ke' :''
, 'serverohon' : 'server mohon'
, 'pas mau' : 'saat'
, 'ketika' : 'saat'
, 'terus2' : 'terus'
, 'teruss' : 'terus'
, 'terusss' : 'terus'
, 'notif' : 'notifikasi'}
```

```

, 'koneksi gagal'          : 'koneksi_gagal'
, 'terjadi kesalahan server' : 'terjadi_kesalahan_server'
, 'loading lama'           : 'loading_lambat'
, 'maksud'                  : ""
, 'maksudnya apa'          : ""
, 'apa maksud'              : ""
, 'apa maksudnya'          : ""
, 'pas'                      : 'saat'
}

co13 = {r'(\b){}(\b)'.format(call13):r'\1{}2'.format(wind13) for call13, wind13 in
kp13.items()}

df411 = df41['Review']=df41['Review'].replace(co13, regex=True)

kp14 = {'yang':"
, 'apaan'          : ""
, 'si'             : 'sih'
, 'sih'            : ""
, 'walaupun'       : 'padahal'
, 'pdhlkan'        : 'padahal'
, 'pdhla'          : 'padahal'
, 'pdhl'           : 'padahal'
, 'pasahal'        : 'padahal'
, 'padhl'          : 'padahal'
, 'sinyal bagus'   : 'sinyal_lancar'
, 'sinyal 4g'       : 'sinyal_lancar'
, 'sinyal 4g penuh' : 'sinyal_lancar'
, 'sinyal ok'       : 'sinyal_lancar'
, 'sinyal penuh'    : 'sinyal_lancar'
, 'sinyal penuh 4g' : 'sinyal_lancar'
, 'sinyal kenceng' : 'sinyal_lancar'

```



```

, 'padahal sinyal_lancar' : 'padahal_sinyal_lancar'
, 'aplikasi apaan kaya gini':"
, 'mohon' : "
, 'maaf' : "
, 'maaff' : "
, 'maaf2' : "
, 'dimaafken' : "
, 'maafnya' : "
, 'sudah di update' : 'padahal sudah di update'
}

```

```

co14 = {r'(\b){ }(\b)'.format(call14):r'\1{ }\2'.format(wind14) for call14, wind14 in
kp14.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co14, regex=True)

```

```

kp15 = { 'padahal sudah di update': 'padahal_sudah_di_update'
, 'warningan': 'warning'
, 'warningnya': 'warning'
, 'alert': 'warning'
, 'di': "
, 'tolong': "
, 'koneksi': 'sinyal'
, 'ngecek': 'cek'
, 'mengecek': 'cek'
, 'pengecekan': 'cek'
, 'neceknnya': 'cek'
, 'mengecekk': 'cek'
, 'ngeceknnya': 'cek'
, 'mengeceknnya': 'cek'
, 'ngeceknnyaaa': 'cek'
, 'ngeceknx': 'cek'
, 'mngecek': 'cek'

```

```

, 'pengeceken': 'cek'
, 'ngecekk': 'cek'
, 'ngecekny': 'cek'
, 'ngecekin': 'cek'
, 'mngecek': 'cek'
, 'ngecek2': 'cek'
, 'pengecekan': 'cek'
, 'pncecekan': 'cek'
, 'sldo': 'saldo'
, 'liat': 'cek'
, 'diliat': 'di cek'
, 'ngeliat': 'cek'
, 'dilihat': 'cek'
, 'lihat': 'cek'
, 'melihat': 'cek'
, 'lihatnya': 'cek'
, 'terlihat': 'cek'
, 'dicek': 'cek'
, 'apl': 'aplikasi'
}

```

```

co15 = {r'(\b){ }(\b)'.format(call15):r'\1{ }\2'.format(wind15) for call15, wind15 in
kp15.items()}

```

```

df411 = df41['Review'] = df41['Review'].replace(co15, regex=True)

```

```

kp16 = {'warning': 'pesan notifikasi_peringatan'
, 'saldonya': 'saldo'
, '0saldonya': 'nol saldo'
, 'saldony': 'saldo'
, 'saldonyo': 'saldo'
, 'teruss': 'terus'
}

```

```

, 'teruss': 'terus'
, 'terussss': 'terus'
, 'terusssss': 'terus'
, 'terussssss': 'terus'
, 'terusssssss': 'terus'
, 'terussssssss': 'terus'
, 'terusssssssss': 'terus'
, 'terussssssssss': 'terus'
, 'terusssssssssss': 'terus'
, 'terussssssssssss': 'terus'
, 'terusssssssssssss': 'terus'
, 'erorterus': 'error terus'
, 'erorterussss': 'error terus'
, 'cek saldo jht': 'cek_saldo_jht'
, 'cek saldo': 'cek_saldo_jht'
}

```

```

co16 = {r'(\b){(\b)'.format(call16):r'\1{\}\2'.format(wind16) for call16, wind16 in
kp16.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co16, regex=True)

```

```

kp17 = {'tidak ada': 'tidak_ada'
, 'tidak ada2': 'tidak_ada'
, 'tidak adanya': 'tidak_ada'
, 'tidak_ada bagusnya': ''
, 'aplikasi sblmnya': 'aplikasi_versi_dulu'
, 'tidak_ada masalah': 'tidak_ada_masalah'
, 'tidak_ada jawaban': 'tidak_ada_respon'
, 'tidak_ada jawabanya': 'tidak_ada_respon'
, 'tidak_ada balasan': 'tidak_ada_respon'
, 'tidak_ada respon': 'tidak_ada_respon'
, 'tidak_ada respond': 'tidak_ada_respon'
, 'tidak_ada response': 'tidak_ada_respon'
}

```

```

, 'tidak_ada response': 'tidak_ada_respon'
, 'tidak_ada respont': 'tidak_ada_respon'
, 'tidak_ada respons': 'tidak_ada_respon'
, 'tidak_ada responnya': 'tidak_ada_respon'
, 'tidak_ada perbaikan': 'tidak_ada_perbaikan'
, 'tidak_ada solusi': 'tidak_ada_solusi'
, 'tidak_ada solusinya': 'tidak_ada_solusi'
, 'tidak_ada nomor antrian': 'tidak_ada_nomor_antrian'
, 'padahal diaplikasi sebelumnya': 'padahal_di aplikasi versi_dulu'

}

co17 = {r'(\b){ }(\b)'.format(call17):r'\1{ }2'.format(wind17) for call17, wind17 in
kp17.items()}

df411 = df41['Review']=df41['Review'].replace(co17, regex=True)

kp18 = {'data saldo tidak_tersedia': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo tidak_tesedia': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo tidak_ada': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'tidak_ada data saldo': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo anda tidak_tesedia': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo anda tidak temukan': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo tak tersedia': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo anda falied': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo selalu tidak_tesedia dan untuk menanbahkan nomor nomor_k_p_j
selalu tidak_bisa isi pin tulisan':"
, 'data saldo not found': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'kenapa saat cek_saldo_jht hanya not found tidak_bisa cek data
saldo': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo tida tersedia': 'muncul_pesan_data_saldo_tidak_tersedia'
, 'data saldo selalu tidak_tesedia': 'muncul_pesan_data_saldo_tidak_tersedia'

```

```
, 'error 404': 'error_404'
, '4gterus': 'lancar terus'
}
```

```
co18 = {r'(\b){ }(\b)'.format(call18):r'\1{ }\2'.format(wind18) for call18, wind18 in
kp18.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co18, regex=True)
```

```
kp19 = { '4g penuh': "
, 'login tapi': 'login_tapi'
, 'knpa': "
, 'knpaaaa': "
, 'knpaarus': "
, 'knpaaa': "
, 'knpaa': "
, 'knpah': "
, 'knpah': "
, 'knpasi': "
, 'kenapa': "
, 'kenapasih': "
, 'kenapaa': "
, 'dahal': "
, 'kenapasi': "
, 'kenapaaaaa': "
, 'kenapah': "
, 'kenapaaaa': "
, 'kenapaj': "
, 'info sebelumnya buka aplikasi pake email sudah daftarkan tidak kenapa2 kenapa
sekarang saat': "
, 'kenapa2': "
, 'kenapaya': "
, 'kenapaya': "

```

```

, 'kenapanya':"
, 'kenapakah':"
, 'kenapaaa':"
, 'kenapasa':"
, 'kenapansih':"
, 'kenapa2':"
, 'ini':"
}

co19 = {r'(\b){}(\b)'.format(call19):r'\1{}\2'.format(wind19) for call19, wind19 in
kp19.items()}

df411 = df41['Review']=df41['Review'].replace(co19, regex=True)

kp20 = {'nanti kalau begini':"
, 'gini':"
, 'begini':"
, 'bgini':"
, 'bgni':"
, 'gni':"
, 'gini2':"
, 'gini':"
, 'bgini2':"
, 'beginian':"
, 'ginian':"
, 'sgni':"
, 'gimana':"
, 'bginian':"
, 'beginilah':"
, 'sengini':"
, 'ginib':"
, 'ginih':"
, 'sgini':"

```

```

, 'seini':"
, 'begini':"
, 'veini':"
, 'sebegini':"
, 'digiini':"
, 'ginikan':"
, 'ginilah':"
, 'kegini':"
, 'beginikah':"
, 'seginikah':"
, 'giliran':"
, 'digiini':"
, 'bigini':"
, 'kayagini':"
}

co20 = {r'(\b){(\b)'.format(call20):r'\1{\2}'.format(wind20) for call20, wind20 in
kp20.items()}

df411 = df41['Review']=df41['Review'].replace(co20, regex=True)

kp21 = { 'kesini':"
, 'inii':"
, 'selamaini':"
, 'disini':"
, 'di sini':"
, 'inih':"
, 'kenapph':"
, 'inipun':"
, 'knp':"
, 'sini':"
, 'inilah':"

```

, 'itulah':"
 , 'alihkan':"
 , 'dialihkan':"
 , 'di alihkan':"
 , 'erroor':'error'
 , 'inilebih':"
 , 'mainin':"
 , 'opini':"
 , 'y':"
 , 'inimah':"
 , 'ksana':"
 , 'ksini':"
 , 'kesana':"
 , 'kesini':"
 , 'ksni':"
 , 'ksna':"
 , 'ke sini':"
 , 'ke sana':"
 , 'ksinii':"
 , 'ksanaa':"
 , 'inii':"
 , 'iiii':"
 , 'iiiiii':"
 , 'iiiiiii':"
 , 'iiiiiiii':"
 , 'iiiiiii':"
 , 'iniblebih':"
 , 'inikah':"
 , 'dsini':"
 , 'inimalah':"
 , 'kini':"
 , 'kisini':"
 , 'inisetelah':"


```

, 'kesinih':"

}

co21 = {r'(\b){ }(\b)'.format(call21):r'\1{ }2'.format(wind21) for call21, wind21 in
kp21.items()}

df411 = df41['Review']=df41['Review'].replace(co21, regex=True)

kp22 = {'login sekali':"
, 'apa gunanya':"
, 'apa':"
, 'tidak_ada_guna':'tidak_ada_guna'
, 'knapa':"
, 'login berhenti':'login_berhenti'
, 'langsung keluar':'force_close'
, 'keluar sendiri':'force_close'
, 'mau':"
, 'gimana':"
, 'tiap':"
, 'gmana':"
, 'begimana':"
, 'gimanalah':"
, 'bagaimana':"
, 'sebagaimana':"
, 'bagaimanakah':"
, 'bagaimanakan':"
, 'bagaimanah':"
, 'bagaimanaaaa':"
, 'bagaimanaaa':"
, 'truss':"
, 'trusss':"
, 'trussss':"

```

```

, 'trusssss':"
, 'trussssss':"
, 'trusssssss':"
, 'trussssssss':"
, 'trusssssssss':"
, 'sudah':"
, 'sdh':"
, 'sudah':"
, 'sdah':"
, 'sesudah':"
, 'sudah_lebih_baik':"
, 'sudahlah':"
, 'sdhlah':"
, 'ah':"
, 'sudhlah':"
, 'sdahlah':"
, 'login':'login'
}

co22 = {r'(\b){(\b)'.format(call22):r'\1{\2}'.format(wind22) for call22, wind22 in
kp22.items()}

df411 = df41['Review']=df41['Review'].replace(co22, regex=True)

kp23 = {'eeerrrooooo': 'error'
, 'dibula': 'dibuka'
, 'di buka': 'dibuka'
, 'dibukak': 'dibuka'
, 'dibukaa': 'dibuka'
, 'dibukaaa': 'dibuka'
, 'dibukaaaa': 'dibuka'
, 'dibukaaaaa': 'dibuka'

```

, 'mulu':"

, 'muluu':"

, 'muluuu':"

, 'muluuuu':"

, 'muluuuuu':"

, 'password':'sandi'

, 'gabisa':'tidak_bisa'

, 'ggabisa':'tidak_bisa'

, 'gbisa':'tidak_bisa'

, 'gabisabisa':'tidak_bisa'

, 'gbisabisa':'tidak_bisa'

, 'nggabisa':'tidak_bisa'

, 'ngga':'tidak'

, 'gabisa2':'tidak_bisa'

, 'gabisaa':'tidak_bisa'

, 'gabisadaftar':'tidak_bisa_daftar'

, 'kgabisa':'tidak_bisa'

, 'gabisaaaa':'tidak_bisa'

, 'ngabisa':'tidak_bisa'

, 'gabisaaa':'tidak_bisa'

, 'kongabisa':'tidak_bisa'

, 'tidak_bisa daftar':'tidak_bisa_daftar'

, 'tidak_bisa daftarr':'tidak_bisa_daftar'

, 'tidak_bisa daftara':'tidak_bisa_daftar'

, 'tidak_bisa daftarkan diri':'tidak_bisa_daftar'

, 'tidak_bisa login':'tidak_bisa_login'

, 'tidak_bisa login2':'tidak_bisa_login'

, 'tidak_bisa logint':'tidak_bisa_login'

, 'tidak terdaftr':'tidak_tradaftar'

, 'tidak terdaftr':'tidak_bisa_login'

, 'tidak_bisa login2':'tidak_bisa_login'

, 'tidak_bisa loginnn':'tidak_bisa_login'

, 'tidak_bisa loginf ':'tidak_bisa_login'

```

, 'bener':'benar'
, 'padahal bener':'benar'
#, 'tidak_bisa dibuka':'tidak_bisa_dibuka'

}

co23 = {r'(\b){ }(\b)'.format(call23):r'\1{\2}'.format(wind23) for call23, wind23 in
kp23.items()}

df411 = df41['Review']=df41['Review'].replace(co23, regex=True)

kp24 = {'benar2':'benar'
, 'bnar':'benar'
, 'benr':'benar'
, 'bnr':'benar'
, 'betul':'benar'
, 'betl':'benar'
, 'btul':'benar'
, 'btl':'benar'
, 'betul2':'benar'
, 'btl2':'benar'
, 'betl2':'benar'
, 'btul2':'benar'
, 'bnar2':'benar'
, 'bnr2':'benar'
, 'segera betulkan':"
, 'lola':'loading_lama'
, 'loading lama':'loading_lama'
, 'segera perbaiki':"
, 'segera diperbaiki':"
, 'segera di perbaiki':"
, 'bpjskarena':'bpjs karena'
, 'karena':"

```

```

, 'disempurnakan':"
, 'di sempurnakan':"
, 'sempurna':"
, 'perbaiki':"
, 'diperbaiki':"
, 'di perbaiki':"
, 'dibenarkan':"
, 'di benarkan':"
, 'dikarenakan':"
, 'karenakan':"
, 'dikarenakn':"
}

co24 = {r'(\b){ }(\b)'.format(call24):r'\1{ }\2'.format(wind24) for call24, wind24 in
kp24.items()}

df411 = df41['Review']=df41['Review'].replace(co24, regex=True)

kp25 = {'blm':'belum'
, 'min':"
, 'admin':"
, 'adminya':"
, 'adminnya':"
, 'mimin':"
, 'miminnya':"
, 'miminnya':"
, 'admint':"
, 'admind':"
, 'admintnya':"
, 'adminn':"
, 'mind':"
, 'adminny':"

```

```

, 'adminnn':"
, 'bantubadmin':"
, 'adminduk':"
, 'administrasi':"
, 'terima kasih':"
, 'terimakasih':"
, 'makasih':"
, 'terima kasihhh':"
, 'perbaikannyaterimakasih':"
, 'terimakasihh':"
, 'bantuan':"
, 'bantuannya':"
, 'gada': 'tidak_ada'

}

co25 = {r'(\b){(\b)'.format(call25):r'\1{\2}'.format(wind25) for call25, wind25 in
kp25.items()}

df411 = df41['Review']=df41['Review'].replace(co25, regex=True)

kp26 = { 'dong':"
, 'tolol':"
, 'dongo':"
, 'nie':"
, 'ngak jelas': 'tidak_jelas'
, 'benerin':"
, 'nih':"
, 'yah':"
, 'klo':"
, 'kalau':"
, 'kalo':"
, 'mlu':"

```



```
, 'nomor_bpjs tidak temukan':'muncul_pesan_nomor_bpjs_tidak_ditemukan'
, 'nomor_bpjs tidak ditemukan':'muncul_pesan_nomor_bpjs_tidak_ditemukan'
, 'nomor_k_p_j tidak temukan':'muncul_pesan_nomor_k_p_j_tidak_ditemukan'
, 'nomor_k_p_j tidak ditemukan':'muncul_pesan_nomor_k_p_j_tidak_ditemukan'
, 'nomor_k_t_p tidak temukan':'muncul_pesan_nomor_k_t_p_tidak_ditemukan'
, 'nomor_k_t_p tidak ditemukan':'muncul_pesan_nomor_k_t_p_tidak_ditemukan'
, 'katp':"
```

```
}
```

```
co26 = {r'(\b){ }(\b)'.format(call26):r'\1{\2}'.format(wind26) for call26, wind26 in
kp26.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co26, regex=True)
```

```
kp27 = {'ni':"
, 'nipu':"
, 'bohong':"
, 'nipuu':"
, 'penipuan':"
, 'penipu':"
, 'pnipu':"
, 'pnpu':"
, 'pnip':"
, 'penpu':"
, 'penip':"
, 'penipu2':"
, 'ok':"
, 'oke':"
, 'okay':"

```


, 'okey':"

, 'anjing':"

, 'jancok':"

, 'patek':"

, 'mbulet': 'susah'

, 'ribet': 'rumit'

, 'ruwet': 'rumit'

, 'opo':"

, 'iki':"

, 'banget':"

, 'bgt':"

, 'bangt':"

, 'bnget':"

, 'bngt':"

, 'lemot': 'lama'

, 'ora': 'tidak'

, 'iso': 'bisa'

, 'isa': 'bisa'

, 'dancuk':"

, 'i':"

, 'mati mati dewe': 'force_close'

, 'mati sendiri': 'force_close'

, 'mbulett': 'susah'

, 'telek':"

, 'cacat':"

, 'cacad':"

, 'cct':"

, 'ccd':"

, 'cact':"

, 'ccat':"

, 'cacad':"

, 'ccad':"

, 'kpu':"

```

, 'ae':"
, 'cok':"
, 'cokk':"
, 'cokkk':"
, 'cokkk':"
, 'cokkkk':"
, 'cokkkkk':"
, 'cokkkkk':"
, 'cokkkkkk':"
, 'cokkkkkkk':"
, 'nomor bpjs':'nomor_bpjs'

}

co27 = {r'(\b){}(\b)'.format(call27):r'\1{}\2'.format(wind27) for call27, wind27 in
kp27.items()}

df411 = df41['Review']=df41['Review'].replace(co27, regex=True)

kp28 = {'masih':"
, 'berat':"
, 'habis ganti':"
, 'masiha':"
, 'masihh':"
, 'terikamasih':"
, 'masihdi':"
, 'sih':"
, 'pajang':"
, 'sij':"
, 'masihhhhh':"
, 'menutup sendiri':'force_close'
, 'tertutup sendiri':'force_close'
, 'ketutup sendiri':'force_close'

```

```

, 'tutup sendiri': 'force_close'
, 'tertutup sendirinya': 'force_close'
, 'logout sendiri': 'force_close'
, 'logout': 'force_close'
, 'tidak': ""
, 'bisa': ""
, 'ada': ""
, 'an': ""
, 'd buka': 'di buka'
, 'internet lancar': 'sinyal_lancar'
, 'kelog out': 'force_close'
, 'internet kencang': 'sinyal_lancar'
, 'internet kencang': 'sinyal_lancar'
, 'loading loading doang': 'loading lama'
, 'loading loading': 'loading lama'
, 'close sendiri': 'force_close'
, 'nomor bpjsx': 'nomor_bpjs'
, 'nomor bpjst': 'nomor_bpjs'
, 'ttl': 'tempat_tanggal_lahir'
, 'tanggal_lahir': 'tempat_tanggal_lahir'
, 'ulang2': 'loading'
, 'apa2an': ""
}

```

```

co28 = {r'(\b){ }(\b)'.format(call28):r'\1{ }2'.format(wind28) for call28, wind28 in
kp28.items()}

```

```

df411 = df41['Review'] = df41['Review'].replace(co28, regex=True)

```

```

kp29 = { 'ndak': 'tidak_bisa'
, 'tk': ""

```

, 'kira':"

, 'ditindaklanjuti':"

, 'akhir':"

, 'akhir2':"

, 'yaa':"

, 'saa':"

, 'karna':"

, 'tp':"

, 'keterangannya':'muncul pesan'

, 'sekarang':"

, 'sekarang':"

, 'lagi':"

, 'lgi':"

, 'lag':"

, 'apalagi':"

, 'lain':"

, 'jeh':"

, 'aplikasi sampah':'aplikasi_tidak_berfungsi'

, 'niat':"

, 'kita':"

, 'ngelag':'loading_lama'

, 'loading lama':'loading_lama'

, 'tok':"

, 'sueee':"

, 'sue':"

, 'suee':"

, 'super':"

, 'masukkan':'memasukan'

, 'kalah sama lama':'aplikasi_versi_dulu'

, 'buang buang waktu':"

, 'buang waktu':"

, 'buang2 waktu':"

, 'buang':"

```

, 'waktu':"
, 'tau tau':'kemudian'
, 'taunya':'kemudian'
, 'tau2':'kemudian'
, 'tambah':"
, 'hrs':"
, 'sgl':"
, 'sie':"
, 'widih':"
, 'widihh':"
, 'widihhh':"
, 'widiih':"
, 'widiiah':"
, 'capek':"
, 'deh':"

}

co29 = {r'(\b){(\b)'.format(call29):r'\1{\2}'.format(wind29) for call29, wind29 in
kp29.items()}

df411 = df41['Review']=df41['Review'].replace(co29, regex=True)

kp30 = {'katanya':"
, 'kata':"
, 'uodate':'update'
, 'awalnya':"
, 'doang':"
, 'macem':"
, 'macam':"
, 'apaah':"
, 'bego':"

```

, 'nambah':"

, 'jelek':"

, 'payah':"

, 'dalam perbaikan': 'pesan_dalam_perbaikan'

, 'claim': 'klaim'

, 'kaga': 'tidak'

, 'gmana':"

, 'cukcuk':"

, 'buset':"

, 'kaya':"

, 'jalanan':"

, 'kocak':"

, 'profesional':"

, 'semakin':"

, 'sinyal bagus': 'sinyal_lancar'

, 'karu karuan':"

, 'masak':"

, 'segera':"

, 'matur nuwun':"

, 'matur':"

, 'nuwun':"

, 'benget':"

, 'itu':"

, 'maksudnya':"

, 'uda':"

, 'd':"

, 'coba':"

, 'kembali':"

, 'biar':"

, 'semua':"

, 'segmen':"

, 'ttp':"

, 'tetap':"

```

, 'tetep':"
, 'cepat':"
, 'cepat':"
, 'gan':"
, 'infox':"
, 'kami':"
, 'para':"
, 'pengangguran':"
, 'butuh':"
, 'uang':"
, 'tabungan':"
, 'jangan':"
, 'dipersulit':"
, 'mbayar':"
}

```

```

co30 = {r'(\b){(\b)'.format(call30):r'\1{\}\2'.format(wind30) for call30, wind30 in
kp30.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co30, regex=True)

```

```

kp31 = {'padahal_sinyal_bagus':'padahal_sinyal_lancar'
, 'waduh':"
, 'kacau':"
, 'nech':"
, 'beda':"
, 'sama':"
, 'versi_lama':'apliaksi_versi_dulu'
, 'msh':"
, 'cara':"
, 'cra':"
, 'car':"
, 'caranya':"

```

, 'hp':"

, 'ram':"

, '2gb':"

, 'terlalu':"

, 'banyak':"

, 'jadi':"

, '2':"

, 'sihh':"

, 'gmna':"

, 'sering':"

, 'request failed':'muncul_pesan_notifikasi_permintaan_gagal'

, 'request time out':'muncul_pesan_notifikasi_permintaan_waktu_habis'

, 'request timeout':'muncul_pesan_notifikasi_permintaan_waktu_habis'

, 'request filled':"

, 'with status code 404':'error_404'

, 'wah':"

, 'tulisan':"

, 'berfungsi':"

, 'penjelasannya':"

, 'request timed out':'muncul_pesan_notifikasi_permintaan_waktu_habis'

, 'setiap':"

, 'kemarin':"

, 'stuck':"

, 'time request':'muncul_pesan_notifikasi_permintaan_waktu_habis'

, 'tulisan':"

, 'punya':"

, 'dengan':"

, 'tku':"

, 'sayangnya':"

, 'telah':"

, 'maunya':"

, 'depan':"

, 'halo':"


```

, 'tanya':"
, 'mengapa':"
, 'sampai':"
, 'bagian':"
, 'selalu':"
, 'solusinya':"
, 'pengiriman':"
, 'request otp':'permintaan otp'
, 'request timeuot':'muncul_pesan_notifikasi_permintaan_waktu_habis'
, 'requested time out':'muncul_pesan_notifikasi_permintaan_waktu_habis'

}

co31 = {r'(\b){}(\b)'.format(call31):r'\1{}\2'.format(wind31) for call31, wind31 in
kp31.items()}

df411 = df41['Review']=df41['Review'].replace(co31, regex=True)

kp32 = {'request filed':"
, 'bilang':"
, 'suruh':"
, 'download':"
, 'donlod':"
, 'donlot':"
, 'eh':"
, 'kan':"
, 'dasar':"
, 'babi':"
, 'kontol':"
, 'potong':"
, 'bulan':"
, 'maksa':"

```

, 'soal':"
, 'kesiapan':"
, 'ribetttt': 'rumit'
, 'skrg':"
, 'koplak':"
, 'w':"
, 'berabad':"
, 'abab':"
, 'abad':"
, 'tiba2':"
, 'log out sndri': 'force_close'
, 'dsruh':"
, 'disruh':"
, 'tpi':"
, 'dioper':"
, 'kemari':"
, 'siap':"
, 'usah':"
, 'bnyk':"
, 'pantesan':"
, 'ikut':"
, 'ternyata':"
, 'seperti':"
, 'diawali':"
, 'sperti':"
, 'yak':"
, 'rp 0': 'nol_rupiah'
, '0 rp': 'nol_rupiah'
, 'atau':"
, 'pengguna':"
, 'korupsi':"
, 'jangan2':"
, 'sumpah':"

```

, 'ngerti':"
, 'harus':"
, 'ngomong':"
, 'gara':"
, 'mempersulit':"
, 'segala':"
, 'tahap':"
, 'pembenaran':"
, 'mungkin':"
, 'mungkin':"
, 'munya':"
, 'sulit':"
, 'rspon':'respon'
, 'resfon':'respon'
, 'narikinya':"
, 'getol':"
, 'ama':'rs'
, 'pro':"
, 'rakyat':"
, 'kecil':"
}

```

```

co32 = {r'(\b){}(\b)'.format(call32):r'\1{}\2'.format(wind32) for call32, wind32 in
kp32.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co32, regex=True)

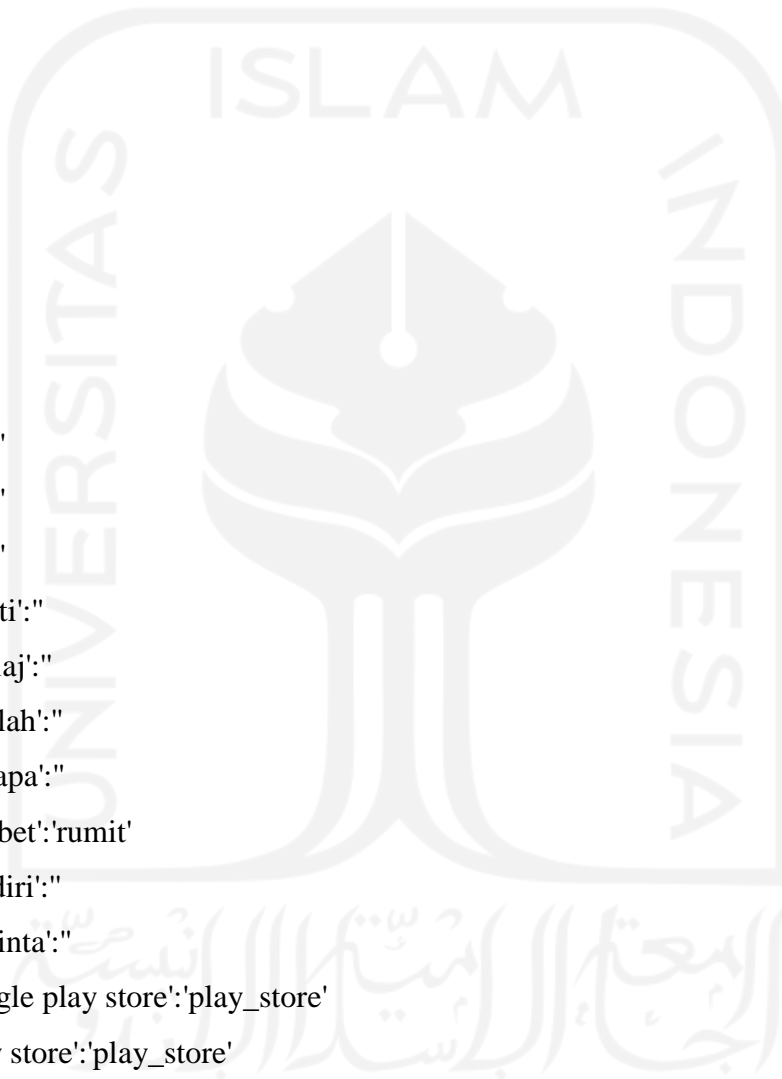
```

```

kp33 = {'kantor':"
, 'drama':"
, 'toh':"
, 'ambil':"
, 'sendiri':"

```

, 'persulit':"
, 'dari':"
, 'senin':"
, 'miris':"
, 'jam':"
, 'mlm':"
, '1':"
, '3':"
, '4':"
, '5':"
, '6':"
, '7':"
, '8':"
, '9':"
, '10':"
, '11':"
, '12':"
, 'mesti':"
, 'okelaj':"
, 'jumlah':"
, 'gapapa':"
, 'peribet': 'rumit'
, 'sendiri':"
, 'diminta':"
, 'google play store': 'play_store'
, 'play store': 'play_store'
, 'awal':"
, 'awal2':"
, 'mkin':"
, 'sehingga':"
, 'asal':"
, 'komen':"
, 'sebel':"



```

, 'heran':"
, 'ngambil':"
, 'duit':"
, 'semoga':"
, 'kena':"
, 'azab':"
, 'suka':"
, 'allah':"
, 'tidur':"

}

co33 = {r'(\b){ }(\b)'.format(call33):r'\1{ }\2'.format(wind33) for call33, wind33 in
kp33.items()}

df411 = df41['Review']=df41['Review'].replace(co33, regex=True)

kp34 = {'berkas':"
, 'serahkan':"
, 'tapi':"
, 'dana':"
, 'belum':"
, 'terima':"
, 'mu':"
, 'koruptor':"
, 'keluhan':"
, 'bukam':"
, 'sengaja':"
, 'motony':"
, 'mudah':"
, 'era':"
, 'digital':"

```

, 'rubah':"
, 'pak':"
, 'jd':"
, 'loh':"
, 'imbasnya':"
, 'ngapain':"
, 'kehambat':"
, 'menghambat':"
, 'maret':"
, 'indonesia':"
, 'bela':"
, 'belain':"
, 'kakek':"
, 'nenek':"
, 'sampai':"
, 'sampe':"
, 'ibu':"
, 'bapak':"
, 'minggu':"
, 'anak muda':"
, 'pada':"
, 'rebutan':"
, 'injek':"
, 'tuh':"
, 'tua':"
, 'lanjutkan':"
, 'mantapp':"
, 'malu':"
, 'malu2in':"
, 'kalah':"
, 'aplikasi buatan swasta':"
, 'buatan':"
, 'swasta':"

```

, 'pemerintah':"
, 'faktor':"
, 'kesengajaan':"
, 'supaya':"
, 'timbul':"
, 'kecurigaan':"
, 'soalnya':"
, 'mana':"
, 'tindak':"
, 'lanjutnya':"

}

co34 = {r'(\b){}(\b)'.format(call34):r'\1{}'\2'.format(wind34) for call34, wind34 in
kp34.items()}

df411 = df41['Review']=df41['Review'].replace(co34, regex=True)

kp35 = {'aplikasi penting nomor sekian':"
, 'sekian':"
, 'penting':"
, 'akh':"
, 'coment':"
, 'bank':"
, 'emang':"
, 'dah':"
, 'tambain':"
, 'bintang':"
, 'lewat':'via'
, 'lwt':'via'
, 'lewt':'via'
, 'lwt':'via'

```

, 'ngerasa':"
, 'digantungin':"
, 'rada':"
, 'soak':"
, 'naikin':"
, 'kapasitas':"
, 'seminggu':"
, 'selasa':"
, 'rabu':"
, 'kamis':"
, 'jumat':"
, 'sabtu':"
, 'minggu':"
, 'sangat':"
, 'dibantu':"
, 'kaka':"
, 'woy':"
, 'dr':"
, 'vc':"
, 'tadi':"
, 'agar':"
, 'lengkap':"
, 'dokumen':"
, 'percepat':"
, 'permudah':"
, 'calo':"
, 'memanfaatka':"
, 'keadaan':"
, 'entah':"
, 'mereka':"
, 'terhubung':"
, 'sistem':"
, 'system':"




```

, 'minta':"
, 'pengajuan':"
, 'ujungnya':"
, 'krn':"
, 'nganggur':"
, 'nunggu':"
, 'bulan':"
, 'blan':"
, 'buln':"
, 'bln':"
, 'skg':"
, 'nungguin':"
, 'nya':"
, 'pny':"
, 'istri':"
, 'suami':"
, 'semudah':"
, 'gaji':"
}

```

```

co35 = {r'(\b){ }(\b)'.format(call35):r'\1{ }\2'.format(wind35) for call35, wind35 in
kp35.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co35, regex=True)

```

```

kp36 = {'rumit rumit':'rumit'

```

```

, 'info':"
, 'hrd':"
, 'luar':"
, 'area':"
, 'tersebut':"
, 'iuran':"
, 'perbulan':"

```

, 'potongan':"
, 'gampang':"
, 'terpotong':"
, 'otomatis':"
, 'untuk':"
, 'covid':"
, 'terphk':"
, 'uangnya':"
, 'utk':"
, 'bertahan':"
, 'hidup':"
, 'keanggotaannya':"
, 'tertolak':"
, 'tlg':"
, 'pandemi':"
, 'bpjs tenaga kerja': 'bpjs'
, 'alasan':"
, 'asli':"
, 'berbayar':"
, 'datang':"
, 'melayani':"
, 'tatap':"
, 'muka':"
, 'tatp':"
, 'ttap':"
, 'mka':"
, 'muk':"
, 'dngn':"
, 'kopad':"
, 'kopid':"
, 'mengembalikan':"
, 'nasabah':"
, 'kecamatan':"

```

, 'tenaga':"
, 'ngurus':"
, 'kelengkapan':"
, 'syarat':"
, 'makan':"
, 'android':"

}

co36 = {r'(\b){}(\b)'.format(call36):r'\1{}\2'.format(wind36) for call36, wind36 in
kp36.items()}

df411 = df41['Review']=df41['Review'].replace(co36, regex=True)

kp37 = {'ngulang':"
, 'apakah':"
, 'tutorial':"
, 'dikonfirmasi':"
, 'sekalian':"
, 'noll':'nol'
, 'taiiiii':"
, '3bln':"
, 'perusahaan':"
, 'tertulis':"
, 'anda':"
, 'perusahaan diputus':"
, 'kerja':"
, 'lg':"
, 'gmn':"
, 'aplikasi aplikasi aplikasi':"
, 'ingin':"
, 'dll':"
, 'enak':"

```

, 'memuaskan':"

, 'dulu aplikasi': 'aplikasi_versi_dulu'

, '100':"

, '50':"

, '60':"

, 'ganti':"

, 'ngaco':"

, 'lu':"

, 'makin':"

, 'percobaan':"

, 'tersedia':"

, 'burukk':"

, 'lebih_baik_lama': 'lebih_baik_aplikasi_versi_dulu'

, 'buruk':"

, 'aplikasi dulu': 'aplikasi_versi_dulu'

, 'kepada':"

, 'pihak':"

, 'pengembang':"

, 'sebagainya':"

, 'baca':"

, 'kritik':"

, 'coustemer':"

, 'pengguna':"

, 'program':"

, 'besar':"

, 'masa':"

, 'kualitas 0':"

, 'kualitas':"

, 'mksh':"

, 'keterangannya':"

, 'tambah':"

, 'nyaman':"

```
}

```

```
co37 = {r'(\b){ }(\b)'.format(call37):r'\1{ }2'.format(wind37) for call37, wind37 in
kp37.items()}

```

```
df411 = df41['Review']=df41['Review'].replace(co37, regex=True)

```

```
kp38 = {'sruh':"
, 'jgn':"
, 'main':"
, 'wei':"
, 'bayar':"
, 'visa':"
, 'account':'akun'
, 'say':"
, 'mastiin':"
, 'parahh':"
, 'tz':"
, 'enakan lama':'lebih_baik_aplikasi_versi_dulu'
, 'payahhhh':"
, 'perbaikannya':"
, 'terbaru':"
, 'percuma':"
, 'mnding lama':'lebih_baik_aplikasi_versi_lama'
, 'lagii':"
, 'seharian':"
, 'paketan kenceng':'internet_lancar'
, 'kagak':'tidak'
, 'taiiii':"
, 'kayanya':"
, 'kudu':"

```

, 'piye':"
, 'mnta':"
, 'satu':"
, 'koq':"
, 'pertanyaanku':"
, 'kayak':"
, 'abal2':"
, 'ge':"
, 'meuni':"
, 'wae':"
, 'ahk':"
, 'brengek':"
, 'dgngn':"
, 'boro':"
, 'balikin aplikasi lama':'lebih_baik_aplikasi_versi_dulu'
, 'lah':"
, 'bla':"
, 'woyyy':"
, 'nyesel':"
, 'dlu':"
, 'aplikasi belum ready':"
, 'ready':"
, 'tak jelas':'tidak_jelas'
, 'tampil bagus':"
, 'terkendala':"
, 'lol':"
, 'serba':"
, 'mempermudah':"
, 'ia':"
, 'terus keluar':'terus force_close'
, 'sedang':"
, 'msk':"
, 'adeh':"

, 'adehh':"

}

co38 = {r'(\b){ }(\b)'.format(call38):r'\1{ }\2'.format(wind38) for call38, wind38 in
kp38.items() }

df411 = df41['Review']=df41['Review'].replace(co38, regex=True)

kp39 = { 'engga': 'tidak'

, 'selisihnya': "

, 'gede': "

, 'juga': "

, 'gitu': "

, 'jaya': "

, 'it': "

, 'maen game moba': "

, 'maen_game_moba': "

, 'kebuka lancar': 'padahal_sinyal_lancar'

, 'kebuka_lancar': 'padahal_sinyal_lancar'

, 'lebih ketimbang aplikasi': 'lebih_baik_aplikasi_versi_lama'

, 'ud': "

, 'sekali': "

, 'malesin': "

, 'lgs': "

, 'aplikasi dulu': 'lebih_baik_aplikasi_versi_dulu'

, 'parah': "

, 'mw': "

, 'perhatikan': "

, 'n': "

, 'nich': "

```

, 'abal':"
, 'hadehh':"
, 'bumn':"
, 'sekelas':"
, 'diomelin':"
, 'balik':"
, 'untung':"
, 'pelayanannya':"
, 'sekiranya':"
, 'nyusahin':"
, 'idiot':"
, 'seandainya':"
, 'koneknya':'koneksi'
, 'bukan bagus':"
, 'acara':"
, 'siatem':"
, 'sgt':"
, 'musah':"
, 'digunakan':"
, 'wtf':"
, 'bobrok':"

```

```

}

```

```

co39 = {r'(\b){ }(\b)'.format(call39):r'\1{ }\2'.format(wind39) for call39, wind39 in
kp39.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co39, regex=True)

```

```

kp40 = {'dikurangin':"
, 'dikit':"

```


, 'micin':"

, 'buat launcing':"

, 'launcing':"

, 'bodoh':"

, 'monyet':"

, 'ngabis2 kuota':"

, 'dibenerin':"

, 'tanggung jawabnya':"

, 'donload':"

, 'tai':"

, 'kbuka':"

, 'sampah':"

, 'gue':"

, 'kampreettttt':"

, 'selse2':'lama'

, 'becus':"

, 'dgn':"

, 'bijak':"

, 'dan benar':"

, 'dan pembaharuan':"

, 'in':"

, 'pembaharuan':"

, 'suru':'suruh'

, 'saat downlod':'saat_download'

, 'downlod':"

, 'gw':"

, 'dg':"

, 'masib lebih_baik dulu':'lebih_baik_aplikasi_versi_dulu'

, 'fiturnya':"

, 'rincian':"

, 'maupun':"

, 'akunx':'akun'

, 'kemaren':"

```

, 'bojs':"
, 'aplikasi close': 'aplikasi force_close'
, 'bos':"
, 'cmn':"
, 'pencitraan':"
, 'hanya':"
, 'mampu':"
, 'menciptakan':"
, 'menampilkan':"
, 'berita':"
, 'dipotong lebih':"
, 'dipotong':"
, 'potong':"
, 'tulisanya':"
, 'penjelasannya':"
, 'bagusnya':"
}

co40 = {r'(\b){(\b)'.format(call40):r'\1{\2}'.format(wind40) for call40, wind40 in
kp40.items()}

df411 = df41['Review']=df41['Review'].replace(co40, regex=True)

kp41 = {'nyoba':"
, 'data2': 'data'
, 'lebih':"
, 'gunakan':"
, 'boss':"
, 'kebobrokan':"
, 'diapus':"
, 'selama':"
, 'tahun':"
, 'diambil':"

```

, 'e klaim': 'e_klaim'
 , 'eklaim': 'e_klaim'
 , 'hadeeeeeh':"
 , 'kendala':"
 , 'a':"
 , 'dua2':"
 , 'apa2':"
 , 'karna':"
 , 'banyak':"
 , 'sarankan':"
 , 'ikuti':"
 , 'demikian':"
 , 'trimakasih':"
 , 'launching':"
 , 'lo':"
 , 'yusahin':"
 , 'ujung':"
 , 'pengembangan':"
 , 'dalam':"
 , 'alart': 'warning'
 , 'bacaan':"
 , 'atasan':"
 , 'kamvret':"
 , 'persetan':"
 , 'bermutu':"
 , 'upgrade':"
 , 'memperburuk':"
 , 'sdg':"
 , 'jwbannya':"
 , 'sampai':"
 , 'kpn':"
 , 'normal':"
 , 'kepastian':"

```

, 'mo':"
, 'suah': 'susah'
, 'aplikasi buat dulu': 'aplikasi_versi_dulu'
, 'kmren':"
, 'nomernya': 'nomor'
, 'memang':"
, 'daripada':"
, 'user':"
, 'friendly':"
, 'aplikasi sebelumnya':"
, 'im':"
, 'dissatisfied':"
, 'terutama':"
, 'karyawan':"
, 'berapa':"
, 'kepotong':"
, 'nic':"
, 'kurang':"
, 'gajinya':"
, 'dikecawain':"
, 'aplikasinya':"
}

```

```

co41 = {r'(\b){}(\b)'.format(call41):r'\1{}\2'.format(wind41) for call41, wind41 in
kp41.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co41, regex=True)

```

```

kp42 = {'luncurin':"
, 'sueeee':"
, 'mening':"
, 'pembaruan':"
, 'mlah':"

```

, 'slalu':"
 , 'skrng':"
 , 'jdi':"
 , 'kesel':"
 , 'tlong':"
 , 'skarang':"
 , 'ngebantu':"
 , 'payahhh':"
 , 'fitur':"
 , 'kecuali':"
 , 'dibilang':"
 , 'tanggapin':"
 , 'follow up':"
 , 'follow':"
 , 'up':"
 , 'diemin':"
 , 'kerjanya':"
 , 'aku':"
 , 'suekkkk':"
 , 'td':"
 , 'jg':"
 , 'aneh':"
 , 'nihh':"
 , 'mentahh':"
 , 'udeh':"
 , 'lama lama':'lama'
 , 'maj':"
 , 'mending bpjs mobile dulu':'lebih_baik_aplikasi_versi_dulu'
 , 'bpjs mobile dulu':'aplikasi_versi_dulu'
 , 'mengecewakan':"
 , 'ucing':"
 , 'sangray':"
 , 'pusing':"

, 'bray':"

, 'x':"

, 'mhh':"

, 'bapet':"

, 'kembangin':"

, 'laennya':"

, 'pembaruan':"

, 'begitu':"

, 'akurat':"

, 'opsinya':"

, 'da':"

, 'danemail':'dan email'

, 'kirim':"

, 'bantuanya':"

, 'tau':"

, 'dpakai':"

, 'mlh':"

, 'srh':"

, 'dounlod':"

, 'bpstku':'bpjs'

, 'ngak':'tidak'

, 'aq':"

, 'dadftar':'daftar'

, 'pembuat':"

, 'aplikasih':'aplikasi'

, 'ksh':"

, 'penjelasan':"

, 'masalah':"

, 'q':"

, 'bagusi':"

, 'aplikasihnya':"

, 'tinggal':"

, 'jancuk':"

```
, 'apanya':"
, 'bru':"
, 'dlm':"
}
```

```
co42 = {r'(\b){ }(\b)'.format(call42):r'\1{ }\2'.format(wind42) for call42, wind42 in
kp42.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co42, regex=True)
```

```
kp43 = {'maintenance':"
, 'sharusnya':"
, 'nd':"
, 'prlu':"
, 'blok':"
, 'tunggu':"
, 'hingga':"
, 'slesai':"
, 'smpurna':"
, 'dblok':"
, 'pngmbangan':"
, 'jdinya':"
, 'memudahkan':"
, 'membuat':'kerepotan'
, 'pun':"
, 'hrus':"
, 'lakukan':"
, 'tsb':"
, 'secara':"
, 'keseluruhan':"
, 'mngkin':"
, 'ulasan':"
, 'tks':"

```

, 'ne':"

, 'woi':"

, 'preeet':"

, 'disuruh':"

, 'sempet':"

, 'dibawa':"

, 'kemana':"

, 'benahi':"

, 'isinya':"

, 'sosmed':"

, 'bingung':"

, 'utamanya':"

, 'iya':"

, 'byar':"

, 'cpet':"

, 'smpe':"

, 'taik':"

, 'tolonglah':"

, 'cepat2':"

, 'pengurusnya':"

, 'berubah':"

, 'dibuat':"

, 'aplikasi sbelumnya lancar':'lebih_baik_aplikasi_versi_dulu'

, 'aj':"

, 'share':"

, 'cari':"

, 'playstore':'play_store'

, 'oknum':"

, 'sedikit':"

, 'dsb':"

, 'dulu mudak praktis':'lebih_baik_apliaksi_versi_dulu'

, 'kayaknya':"

, 'lma':'lama'


```

, 'karepmu':"
, 'opooooo':"
, 'ceeeee':"
, 'mangkelno':"
, 'asliiiii':"
, 'dech':"

}

co43 = {r'(\b){ }(\b)'.format(call43):r'\1{ }\2'.format(wind43) for call43, wind43 in
kp43.items()}

df411 = df41['Review']=df41['Review'].replace(co43, regex=True)

kp44 = {'kmrn':"
, 'baik2':"
, 'masi':"
, 'luncurkan':"
, 'hari':"
, 'lalu':"
, 'bermanfaat':"
, 'pake':'pakai'
, 'kecewa':"
, 'nformasinya':"
, 'trima kasih':"
, 'menerus':"
, 'perubahan':"
, 'dama':"
, 'ampun':"
, 'benrrin':"
, 'bangke':"
, 'kedaftar':"

```

, 'pke': 'pakai'
, 'dulu2': 'dulu'
, 'akhirnya': "
, 'diperbaharui': "
, 'msih': "
, 'jempol': "
, 'wkwkwk': "
, 'pbaruan': "
, 'hak': "
, 'selaku': "
, 'anggota': "
, 'somprrrrrr': "
, 'sendang': "
, 'hank': "
, 'koment': "
, 'dishare': "
, 'menggunakan': "
, 'stlh': "
, 'membingungkan': "
, 'sd': "
, 'milik': "
, 'negara': "
, 'sungguh': "
, 'duitny': "
, 'dikorup': "
, 'pegawainy': "
, 'makany': "
, 'akses': "
, 'alangkah': "
, 'lucuny': "
, 'ngabisin': "
, 'kuota': "
, 'sistemnya': "



```

, 'down':"
, 'statusnya':"
, 'kya':"
, 'pantat':"
, 'berfikir':"
, 'matang':"
, 'hadeuh':"
, 'cape':"
, 'deeeeh':"
, 'pdahl':'padahal'
, 'ssuai':"
, 'katany':"
, 'trdaftar':'terdaftar'
}

```

```

co44 = {r'(\b){}(\b)'.format(call44):r'\1{}\2'.format(wind44) for call44, wind44 in
kp44.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co44, regex=True)

```

```

kp45 = {'memuakkan':"
, 'berminggu':"
, 'pikir':"
, 'perbaikan':"
, 'nyatanya':"
, 'emg':"
, 'pasti':"
, 'masah':"
, 'giman':"
, 'alasanya':"
, 'sblm':"
, 'diperbarui':"

```

, 'payahhhhhhhhhhh':"
 , 'ngawur':"
 , 'gnti': 'ganti'
 , 'sm':"
 , 'gimna':"
 , 'sch':"
 , 'payahhhhhhhhhhh':"
 , 'yh':"
 , 'ngasih':"
 , 'masukn':"
 , 'direspon':"
 , 'sumpah':"
 , 'sumpahh':"
 , 'tiba':"
 , 'dn':"
 , 'karu':"
 , 'gara2':"
 , 'telplah':"
 , 'telp':"
 , 'wis':"
 , 'jal':"
 , 'jelas':"
 , 'pabrik':"
 , 'sekarng':"
 , 'jls':"
 , 'katnya':"
 , 'moe':"
 , 'keselin':"
 , 'nh':"
 , 'salah':"
 , 'ngirim':"
 , 'balesan':"
 , 'berkali':"

```

, 'kali':"
, 'slah':"
, 'profile':"
, 'haduuhh':"
, 'too':"
, 'booossss':"
, 'haduh':"
, 'pd':"
, 'rinci':"
, 'kug':"
, 'kamu':"
, 'sllu':"
, 'sehat':"
, 'lindungi':"
, 'tuhan':"
, 'menyulitkan':"
, 'saat masuk': 'saat_memasukkan'
, 'saat masukin': 'saat_memasukkan'
, 'saat masukan': 'saat_memasukkan'

}

```

```

co45 = {r'(\b){ }(\b)'.format(call45):r'\1{ }\2'.format(wind45) for call45, wind45 in
kp45.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co45, regex=True)

```

```

kp46 = {'apaaan':"
, 'tdi':"
, 'kebuka':"
, 'terdftar': 'terdaftar'
, 'buka':"

```

```

, 'nyuruh':"
, 'dooooong':"
, 'secepatnya':"
, 'trim':"
, 's':"
, 'malah':"
, 'mikir':"
, 'sebelumnya':"
, 'tq':"
, '2g 3g 4g 5g 6g 7g':"
, 'saat masuk akun':'saat_masuk_akun'
, 'telefon':'telepon'
, 'telpon':'telepon'

}

```

```

co46 = {r'(\b){(\b)'.format(call46):r'\1{\2}'.format(wind46) for call46, wind46 in
kp46.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co46, regex=True)

```

```

kp47 = { 'baru':"
, 'barunya':"
, 'terbaru':"
, 'trbaru':"
, 'perbarui':"
, 'msa':"
, 'kang':"
, 'tohh':"
, 'terbarunya':"
, 'beres':"
, 'juganot':'juga not'

```

```

, 'not found': 'pesan_notif_tidak_ditemukan'
, 'bagus': ''
, 'emailnya': 'email'
, 'emai': 'email'
, 'paswd': 'sandi'

}

co47 = {r'(\b){ }(\b)'.format(call47):r'\1{ }2'.format(wind47) for call47, wind47 in
kp47.items()}

df411 = df41['Review']=df41['Review'].replace(co47, regex=True)

kp48 = {'nx': ''
, 'ad': ''
, 'mendaftar': ''
, 'mendaftarkan': ''
, 'mendaftarknyah': ''
, 'mendaftarnya': ''
, 'mendaftarkn': ''
, 'kasih': ''
, 'aplikasi aplikasi': 'aplikasi'
, 'diaplikasi': ''
, 'buat akun': 'buat_akun'
, 'bpjs ketenagakerjaan': ''
, 'bpjs ketenagakerjaannya': ''
, 'almt': 'alamat'

}

co48 = {r'(\b){ }(\b)'.format(call48):r'\1{ }2'.format(wind48) for call48, wind48 in
kp48.items()}

```

```
df411 = df41['Review']=df41['Review'].replace(co48, regex=True)
```

```
kp49 = {'ketenagakerjaan':"
    , 'ketenagakerjaannya':"
    , 'bpjs ketenagakerjaanku':"
    , 'ketenagakerjaanku':"
    , 'bpjs ketenagakerjaankan':"
    , 'cek pakai':'cek_pakai'
    , 'pakai versi':'pakai_aplikasi_versi_dulu'
    , 'pakai aplikasi_versi_dulu':'pakai_aplikasi_versi_dulu'
    , 'model lama':'aplikasi_versi_dulu'
    , 'alamat email_yang_dulu':'email_yang_dulu'
    , 'alamat akun':'email'
    , 'alamat emailny':'email'
    , 'alamat emailku':'email'
    , 'e mail':'email'
    , 'e mailnya':'email'
    , 'google mail':'email'
    , 'mail':'email'
    , 'alamat email':'email'
    , 'bpjs online':'bpjs'
    , 'bpjs onlinenya':"
    , 'nomor_kartu_bpjs online':'nomor_kartu_bpj'
    , 'susah buat login':'susah_login'
    , 'susah buat login_email':'susah_login_email'

}
```

```
co49 = {r'(\b){ }(\b)'.format(call49):r'\1 {}2'.format(wind49) for call49, wind49 in
kp49.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co49, regex=True)
```



```

kp50 = {'susah buat cek_saldo_jht':'susah_cek_saldo_jht'
, 'susah buat_akun':'susah_buat_akun'
, 'susah dibuka':'susah_dibuka'
, 'susah dibukanya':'susah_dibuka'
, 'susahnya login':'susah_login'
, 'susah login':'susah_login'
, 'susah loginnya':'susah_login'
, 'susah loginya':'susah_login'
, 'susah login_email tidak_terdaftar':'login_email tidak_terdaftar'
, 'susah loginn':'susah_login'
, 'login susah':'susah_login'
, 'login susahnya':'susah_login'
, 'susah klaim':'susah_klaim'
, 'susah daftar':'susah_daftar'
, 'susah daftarin':'susah_daftar'
, 'susah daftarnya':'susah_daftar'
, 'cek susah':'cek_susah'
, 'susah cek_saldo_jht':'susah_cek_saldo_jht'
, 'susah cek':'susah_cek'
, 'aplikasi susah':'aplikasi_susah'
, 'daftar susahnya':'daftar_susah'
, 'daftar susah':'daftar_susah'
, 'telephone':'telepon'
}

```

```

co50 = {r'(\b){ }(\b)'.format(call50):r'\1 { } \2'.format(wind50) for call50, wind50 in
kp50.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co50, regex=True)

```

```

kp51 = {'susah masuk':'susah_masuk'
, 'susah masuknya':'susah_masuk'
, 'susah masukan':'susah_masuk'
, 'susah masukannya':'susah_masuk'
, 'susah masukin':'susah_masuk'
, 'susah masukny':'susah_masuk'
, 'susah masukkk':'susah_masuk'
, 'susah masukknya':'susah_masuk'
, 'aplikasi_susah masuk':'aplikasi_susah_masuk'
, 'aplikasi_susah masuknya':'aplikasi_susah_masuk'
, 'daftar_susah masukk':'daftar_susah'
, 'daftar_susah masuk':'daftar_susah'
, 'masuk susah':'susah_masuk'
, 'klaim susah':'klaim_susah'
, 'e_klaim susah':'e_klaim_susah'
, 'diklaim susah':'klaim_susah'
, 'susah buat dibuka':'susah_buat_dibuka'
, 'susah buat dibukanya':'susah_buat_dibuka'
, 'buat login':'login'
, 'login buat':'login'
, 'buat':''
, 'nomor telepon':'nomor_telepon'
, 'email_terdaftar':'email_terdaftar'
}

```

```

co51 = {r'(\b){ }(\b)'.format(call51):r'\1{ }2'.format(wind51) for call51, wind51 in
kp51.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co51, regex=True)

```

```

kp52 = {'nomor teleponnya':'nomor_telepon'
        , 'nomor kode':'nomor_kode'
        , 'nomor pin':'nomor_pin'
        , 'nomor pinya':'nomor_pin'
        , 'nomor pinnya':'nomor_pin'
        , 'nomor kepesertaan':'nomor_kepesertaan'
        , 'nomor peserta':'nomor_peserta'
        , 'nomor pesertanya':'nomor_peserta'
        , 'nomor antrian':'nomor_antrian'
        , 'nomor antrianya':'nomor_antrian'
        , 'nomor antriany':'nomor_antrian'
        , 'nomor antrianpun':'nomor_antrian'
        , 'nomor antriannya':'nomor_antrian'
        , 'email terdaftar':'email_terdaftar'
        , 'padahalemail terdaftar':'email_terdaftar'
        , 'email belum terdaftar':'email_belum_terdaftar'
        , 'email terdaftar':'email_terdaftar'
    }

co52 = {r'(\b){ }(\b)'.format(call52):r'\1{ }2'.format(wind52) for call52, wind52 in
kp52.items()}

df411 = df41['Review']=df41['Review'].replace(co52, regex=True)

kp53 = {'email terdaftar':'email_terdaftar'
        , 'login_email terdaftar':'login_email_tidak_terdaftar'
        , 'login_email terdaftar':'login_email_tidak_terdaftar'

```

```

, 'login_email terdaftar':'login_email_tidak_terdaftar'
, 'login_email terdaftar':'login_email_tidak_terdaftar'
, 'klaim susah':'klaim_susah'
, 'klaim susahnya':'klaim_susah'
, 'e_klaim susah':'e_klaim_susah'
, 'klaim susah':'klaim_susah'
, 'klaim susahnya':'klaim_susah'
, 'klaim susah':'klaim_susah'
, 'ngklaim susahnya':'klaim_susah'
, 'klaim susahnya':'klaim_susah'
, 'mengklaim susah':'klaim_susah'
, 'diklaim susah':'klaim_susah'
, 'klaim susahny':'klaim_susah'
, 'mengklaim susah':'klaim_susah'
, 'ngeklaim susah':'klaim_susah'
, 'e_klaim susahnya':'klaim_susah'
, 'susah_masuk loading_nya':'susah masuk_loading'
, 'susah_masuk loading_lama':'susah masuk_loading_lama'
, 'susah klaim':'klaim_susah'
, 'susah klaim':'klaim_susah'
, 'susah klaim':'klaim_susah'
, 'persusah klaim':'klaim_susah'
}

```

```

co53 = {r'(\b){ }(\b)'.format(call53):r'\1{ }2'.format(wind53) for call53, wind53 in
kp53.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co53, regex=True)

```

```

kp54 = {'susah klaim':'klaim_susah'
, 'email terdaftar':'email_terdaftar'
, 'email terdaftar':'email_terdaftar'

```

```
, 'email    terdaftar':'email_terdaftar'  
, 'email tidak terdaftar':'email_tidak_terdaftar'  
, 'email    tidak terdaftar':'email_tidak_terdaftar'  
, 'keterangan':'muncul_pesan'  
, 'keteranganya':'muncul_pesan'  
, 'keteranganx':'muncul_pesan'  
, 'diketerangannya':'muncul_pesan'  
, 'keterangany':'muncul_pesan'  
, 'keterangannga':'muncul_pesan'  
, 'keteranganne':'muncul_pesan'  
, 'keterangang':'muncul_pesan'  
, 'error code 404':'error_404'  
, 'error 404':'error_404'  
, 'bug':'error'  
, 'bugs':'error'  
, 'ngebug':'error'  
, 'bugsnya':'error'  
, 'debug':'error'  
, 'bugnya':'error'  
, 'bugggg':'error'  
, 'bugg':'error'  
, 'bugsnya':'error'  
, 'buga':'error'  
, 'bugkah':'error'  
, 'bug2':'error'  
, 'ngbugg':'error'  
, 'buggggg':'error'  
, 'ngebugg':'error'  
, 'errorrrrr':'error'  
, 'errorr':'error'  
, 'error2':'error'  
, 'errornya':'error'  
}
```

```
co54 = {r'(\b){ }(\b)'.format(call54):r'\1 { }\2'.format(wind54) for call54, wind54 in
kp54.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co54, regex=True)
```

```
kp55 = {'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'aplikasi error':'aplikasi_error'
, 'login error':'login_error'
, 'login error':'login_error'
, 'login error':'login_error'
, 'login error':'login_error'
, 'login error':'login_error'
, 'belum':"
, 'aplikasi sebelumnya':'aplikasi_versi_dulu'
, 'tidak bisa cek':'tidak_bisa_cek'
, 'aplikasi masuk':'aplikasi_masuk'
, 'aplikasi masuk':'aplikasi_masuk'
, 'aplikasi masuk':'aplikasi_masuk'
, 'aplikasi masukan':'aplikasi_memasukkan'
, 'aplikasi masukan':'aplikasi_memasukkan'
, 'aplikasi masuk2':'aplikasi_masuk'
, 'aplikasi masukny':'aplikasi_masuk'
, 'aplikasi masukan':'aplikasi_memasukkan'
, 'aplikasi masukan':'aplikasi_memasukkan'
, 'aplikasi masuk2':'aplikasi_masuk' }
```

```

, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
, 'aplikasi masuk': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masuk': 'aplikasi_masuk'
, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
, 'aplikasi masuk': 'aplikasi_masuk'
, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
, 'aplikasi masuk': 'aplikasi_masuk'
, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
, 'aplikasi masuk': 'aplikasi_masuk'
, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
, 'aplikasi masuk': 'aplikasi_masuk'
, 'aplikasi masukny': 'aplikasi_masuk'
, 'aplikasi masukin': 'aplikasi_memasukkan'
, 'aplikasi masukan': 'aplikasi_memasukkan'
}

```

```
co55 = {r'(\b){ }(\b)'.format(call55):r'\1 { }\2'.format(wind55) for call55, wind55 in
kp55.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co55, regex=True)
```

```
kp56 = {'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'aplikasi aplikasi':'aplikasi'
, 'trus':"
, 'terus':"
, 'terus2an':"
, 'seterusnya':"
, 'terussa':"
, 'terusan':"
, 'diteruskan':"
, 'teruskan':"
, 'terusnya':"
, 'terusane':"
, 'teruslah':"
, 'terussaa':"
, 'terusdr':"
, 'terusa':"
, 'terusin':"
, 'terusterusan':" }
```



```

, 'jaringan error': 'jaringan_error'
, 'jaringan error': 'jaringan_error'
, 'jaringan error': 'jaringan_error'
, 'jaringan error': 'jaringan_error'
, 'install': ""
, 'diinstall': ""
, 'uninstall': ""
, 'uninstalled': ""
, 'terinstall': ""
, 'terinstall': ""
, 'uinstall': ""
, 'instal': ""
, 'reinstall': ""
, 'cba': ""
, 'diuninstall': ""
, 'installl': ""
, 'uninstallbpjstku': ""
, 'installagi': ""
, 'uninstal': ""
, 'uninstallll': ""
, 'diuninstall': ""
, 'installation': ""
, 'menginstall': ""
, 'dinstall': ""
, 'menginstall': ""
, 'installed': ""
, 'desember': ""
, 'jan': ""
, '2020': ""
}

```

```

co57 = {r'(\b){ }(\b)'.format(call57):r'\1{ }2'.format(wind57) for call57, wind57 in
kp57.items()}

```



```
co58 = {r'(\b){ }(\b)'.format(call58):r'\1{\2}'.format(wind58) for call58, wind58 in
kp58.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co58, regex=True)
```

```
kp59 = {'aplikasi guna':"
, 'aplikasi gunanya':"
, 'aplikasi guna':"
, 'aplikasi gunaa':"
, 'aplikasi gunanya':"
, 'aplikasi gunaaa':"
, 'aplikasi guna':"
, 'aplikasi gunainnya':"
, 'aplikasi guna':"
, 'aplikasi guna':"
, 'aplikasi gunain':"
, 'aplikasi gunanya':"
, 'aplikasi gunain':"
, 'aplikasi guna':"
, 'aplikasi gunaa':"
, 'aplikasi daftar':'daftar'
, 'aplikasi daftar':'daftar'
, 'aplikasi daftarnya':'daftar'
, 'aplikasi daftarnya':'daftar'
, 'aplikasi daftar':'daftar'
, 'aplikasi daftarin':'daftar'
, 'aplikasi daftarkan':'daftar'
, 'aplikasi daftar_susah':'daftar_susah'
, 'aplikasi daftar':'daftar'
, 'aplikasi daftar':'daftar'
, 'aplikasi daftarkan':'daftar'
, 'aplikasi daftarin':'daftar' }
```

```

, 'aplikasi    daftar': 'daftar'
, 'aplikasi    daftar_susah': 'daftar_susah'
, 'aplikasi    daftar': 'daftar'
, 'aplikasi    daftar': 'daftar'
, 'aplikasi    daftar': 'daftar'
, 'aplikasi    daftar': 'daftar'
, 'aplikasi    daftar': 'daftar'
, 'closeapp': 'force_close'
}

```

```

co59 = {r'(\b){ }(\b)'.format(call59):r'\1{ }2'.format(wind59) for call59, wind59 in
kp59.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co59, regex=True)

```

```

kp60 = { 'bpjs mobile': "
, 'login susah': 'susah_login'
, 'login susah': 'susah_login'
, 'login susah': 'susah_login'
, 'login susah': 'susah_login'
, 'login susah': 'susah_login'
, 'login susah': 'susah_login'
, 'login susah': 'susah_login'
, 'aplikasi susah_dibuka': 'aplikasi_tidak_bisa_dibuka'
, 'aplikasi susah_dibuka': 'aplikasi_tidak_bisa_dibuka'
, 'aplikasi susah_dibuka': 'aplikasi_tidak_bisa_dibuka'
, 'aplikasi susah_dibuka': 'aplikasi_tidak_bisa_dibuka'
, 'login tidak_bisa': 'tidak_bisa_login'
, 'dilogin tidak_bisa': 'tidak_bisa_login'
, 'login tidak_bisa': 'tidak_bisa_login'
, 'login tidak_bisa': 'tidak_bisa_login'
, 'setelah': "

```



```
, 'login_email    terdaftar':'login_email_tidak_terdaftar'
, 'login_email    terdaftar':'login_email_tidak_terdaftar'

}
```

```
co62 = {r'(\b){ }(\b)'.format(call62):r'\1{ } \2'.format(wind62) for call62, wind62 in
kp62.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co62, regex=True)
```

```
kp62 = { 'terjadi kesalahan':'terjadi_kesalahan'
, 'sedang terjadi kesalahan':'sedang terjadi_kesalahan'
, 'terjadi kesalahann':'terjadi_kesalahan'
, 'terjadi kesalahannn':'terjadi_kesalahan'
, 'terjadi kesalahan':'tarjadi_kesalahan'
, 'kesalahan terjadi':'tarjadi_kesalahan'
, 'kesalahan terjadi':'terjadi_kesalahan'
, 'dancok':"
}
```

```
co62 = {r'(\b){ }(\b)'.format(call62):r'\1{ } \2'.format(wind62) for call62, wind62 in
kp62.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co62, regex=True)
```

```
kp63 = { 'terjadi_kesalahan terjadi_kesalahan':'terjadi_kesalahan'
, 'trs':"
, 'aplikasi berhenti':'aplikasi_berhenti'
, 'aplikasi berhenti':'aplikasi_berhenti'
, 'aplikasi berhentikan':'aplikasi_berhenti'
, 'aplikasi berhenti':'aplikasi_berhenti' }
```



```

, 'penuh antrian online':"
, 'penuh antrian online':"
, 'penuh antrian online':"
, 'penuh antrian online':"
, 'penuh antrian online':"
, 'penuh antrian online':"
, 'penuh antrian online':"
, 'penuh antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'antrian online':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"
, 'online antrian':"

```

```

}

```

```

co64 = {r'(\b){ }(\b)'.format(call64):r'\1{ }2'.format(wind64) for call64, wind64 in
kp64.items()}

```



```
co65 = {r'(\b){ }(\b)'.format(call65):r'\1{\}\2'.format(wind65) for call65, wind65 in
kp65.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co65, regex=True)
```

```
kp65 = { 'tidak_bisa masuk':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk ':'tidak_bisa_masuk'
, 'tidak_bisa masuk2':'tidak_bisa_masuk'
, 'tidak_bisa masukkk':'tidak_bisa_masuk'
, 'tidak_bisa masuklah':'tidak_bisa_masuk'
, 'tidak_bisa masuk2':'tidak_bisa_masuk'
, 'tidak_bisa masukin':'tidak_bisa_memasukkan'
, 'tidak_bisa masukk':'tidak_bisa_masuk'
, 'tidak_bisa masukkkk':'tidak_bisa_masuk'
, 'tidak_bisa masukk2':'tidak_bisa_masuk'
, 'tidak_bisa masukkkkk':'tidak_bisa_masuk'
, 'tidak_bisa masukkkkmm':'tidak_bisa_masuk'
, 'tidak_bisa masukmasuk':'tidak_bisa_masuk'
, 'tidak_bisa masukan':'tidak_bisa_memasukkan'
, 'tidak_bisa masukmasukkk':'tidak_bisa_masuk'
, 'tidak_bisa masukmasuk':'tidak_bisa_masuk'
, 'tidak_bisa masukkin':'tidak_bisa_memasukkan'

, 'tidak_bisa masuk2':'tidak_bisa_masuk'
, 'tidak_bisa masukkk':'tidak_bisa_masuk'
, 'tidak_bisa masuklah':'tidak_bisa_masuk'
, 'tidak_bisa masuk2':'tidak_bisa_masuk' }
```

, 'tidak_bisa masukin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masukk': 'tidak_bisa_masuk'

, 'tidak_bisa masukkkk': 'tidak_bisa_masuk'

, 'tidak_bisa masukk2': 'tidak_bisa_masuk'

, 'tidak_bisa masukkkkk': 'tidak_bisa_masuk'

, 'tidak_bisa masukkkkmm': 'tidak_bisa_masuk'

, 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'

, 'tidak_bisa masukan': 'tidak_bisa_memasukkan'

, 'tidak_bisa masukmasukkk': 'tidak_bisa_masuk'

, 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'

, 'tidak_bisa masukkin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masuk2': 'tidak_bisa_masuk'

, 'tidak_bisa masukkk': 'tidak_bisa_masuk'

, 'tidak_bisa masuklah': 'tidak_bisa_masuk'

, 'tidak_bisa masuk2': 'tidak_bisa_masuk'

, 'tidak_bisa masukin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masukk': 'tidak_bisa_masuk'

, 'tidak_bisa masukkkk': 'tidak_bisa_masuk'

, 'tidak_bisa masukk2': 'tidak_bisa_masuk'

, 'tidak_bisa masukkkkkk': 'tidak_bisa_masuk'

, 'tidak_bisa masukkkkmm': 'tidak_bisa_masuk'

, 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'

, 'tidak_bisa masukan': 'tidak_bisa_memasukkan'

, 'tidak_bisa masukmasukkk': 'tidak_bisa_masuk'

, 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'

, 'tidak_bisa masukkin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masuk2': 'tidak_bisa_masuk'

, 'tidak_bisa masukkk': 'tidak_bisa_masuk'

, 'tidak_bisa masuklah': 'tidak_bisa_masuk'

, 'tidak_bisa masuk2': 'tidak_bisa_masuk'

, 'tidak_bisa masukin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masukk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukk2': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkkkkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkkkmm': 'tidak_bisa_masuk'
 , 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukan': 'tidak_bisa_memasukkan'
 , 'tidak_bisa masukmasukkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masuk2': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masuklah': 'tidak_bisa_masuk'
 , 'tidak_bisa masuk2': 'tidak_bisa_masuk'
 , 'tidak_bisa masukin': 'tidak_bisa_memasukkan'
 , 'tidak_bisa masukk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukk2': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkkkkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkkkmm': 'tidak_bisa_masuk'
 , 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukan': 'tidak_bisa_memasukkan'
 , 'tidak_bisa masukmasukkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukmasuk': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkin': 'tidak_bisa_memasukkan'

, 'tidak_bisa masuk2': 'tidak_bisa_masuk'
 , 'tidak_bisa masukkk': 'tidak_bisa_masuk'
 , 'tidak_bisa masuklah': 'tidak_bisa_masuk'
 , 'tidak_bisa masuk2': 'tidak_bisa_masuk'
 , 'tidak_bisa masukin': 'tidak_bisa_memasukkan'
 , 'tidak_bisa masukk': 'tidak_bisa_masuk'

, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar': 'force_close'
, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'
, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'
, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'
, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'
, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'

```
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'

, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi_force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'

, 'masuk keluar2': 'force_close'
, 'aplikasi_masuk keluar': 'aplikasi_force_close'
, 'tidak_bisa_masuk keluar aplikasi': 'tidak_bisa_masuk aplikasi_force_close'
, 'tidak_bisa_masuk keluar': 'tidak_bisa_masuk aplikasi_force_close'

}
```

```
co66 = {r'(\b){ }(\b)'.format(call66):r'\1{ }2'.format(wind66) for call66, wind66 in
kp66.items() }
```

```
df411 = df41['Review'] = df41['Review'].replace(co66, regex=True)
```

```
kp67 = { 'masukin': 'memasukkan'
, 'masukin': 'memasukkan'

, 'dimasukin': 'memasukkan'
, 'nomasukin': 'memasukkan'
, 'masukinnya': 'memasukkan'
, 'masukinnya': 'memasukkan'
, 'dmasukin': 'memasukkan'
, 'masukinny': 'memasukkan'
, 'msukin': 'memasukkan'
, 'masukinya': 'memasukkan'

, 'dimasukan': 'memasukkan'
, 'nomasukan': 'memasukkan' }
```

```

, 'masukannya':'memasukkan'
, 'masukannya':'memasukkan'
, 'dmasukan':'memasukkan'
, 'masukanny':'memasukkan'
, 'msukan':'memasukkan'
, 'masukanya':'memasukkan'
, 'memasukan':'memasukkan'
, 'semasukan':'memasukkan'
, 'pemasukan':'memasukkan'
, 'memasukanya':'memasukkan'
, 'dipemasukan':'memasukkan'
, 'masukann':'memasukkan'
, 'masukandata':'memasukkan data'
, 'masukanyq':"

, 'woi':"
, 'woii':"
, 'woiii':"
, 'woiiii':"
, 'woiiiii':"
, 'woiiiiii':"
, 'woiiiiiii':"
, 'woiiiiiiii':"
, 'woiiiiiii':"
, 'woiiiiiiii':"

, 'handphone':'telepon'
, 'smartphone':'telepon'

}

```

```

co67 = {r'(\b){(\b)'.format(call67):r'\1{\2}'.format(wind67) for call67, wind67 in
kp67.items()}

```


, 'saat memasukkan_nomor_k_p_j': 'saat_memasukkan_nomor_k_p_j'

, 'saat memasukkan_nomor_telepon': 'saat_memasukkan_nomor_telepon'

, 'saat memasukkan_nomor_kartu_bpjs': 'saat_memasukkan_nomor_kartu_bpjs'

, 'saat memasukkan_nomor_k_t_p': 'saat_memasukkan_nomor_k_t_p'

, 'saat memasukkan_nomor_k_p_j': 'saat_memasukkan_nomor_k_p_j'

, 'saat memasukkan_nomor_telepon': 'saat_memasukkan_nomor_telepon'

, 'saat memasukkan_nomor_kartu_bpjs': 'saat_memasukkan_nomor_kartu_bpjs'

, 'saat memasukkan_nomor_k_t_p': 'saat_memasukkan_nomor_k_t_p'

, 'saat memasukkan_nomor_k_p_j': 'saat_memasukkan_nomor_k_p_j'

, 'saat memasukkan_nomor_telepon': 'saat_memasukkan_nomor_telepon'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'saat memasukkan': 'saat_memasukkan'

, 'disaat memasukkan_tempat_tanggal_lahir': 'saat memasukkan_tempat_tanggal_lahir'

, 'saat memasukkan_tempat_tanggal_lahir': 'saat_memasukkan_tempat_tanggal_lahir'

, 'disaat memasukkan_tempat_tanggal_lahir': 'saat memasukkan_tempat_tanggal_lahir'

, 'saat memasukkan_tempat_tanggal_lahir': 'saat_memasukkan_tempat_tanggal_lahir'

, 'disaat memasukkan_tempat_tanggal_lahir': 'saat memasukkan_tempat_tanggal_lahir'


```

, 'disaat    memasukkan':'saat_memasukkan'
, 'disaat    memasukkan':'saat_memasukkan'
, 'disaat    memasukkan':'saat_memasukkan'
, 'disaat    memasukkan':'saat_memasukkan'

, 'errorrrrrr':'error'
}

co69 = {r'(\b){ }(\b)'.format(call69):r'\1{ }\2'.format(wind69) for call69, wind69 in
kp69.items()}

df411 = df41['Review']=df41['Review'].replace(co69, regex=True)

kp70 = {'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'saat_memasukkan nomor':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'nomor memasukkan':'saat_memasukkan_nomor'
, 'padahal benar':'padahal_benar'
, 'padahal benar':'padahal_benar'
, 'padahal benar':'padahal_benar'
, 'padahal benar':'padahal_benar'
, 'padahal benar':'padahal_benar'

```


, 'kode verifikasiny':'kode_verifikasi'
, 'kode verifikasix':'kode_verifikasi'
, 'kode verifikasin':'kode_verifikasi'
, 'kode verifikasihnya':'kode_verifikasi'
, 'kode verifikasinya':'kode_verifikasi'
, 'kode verifikasiny':'kode_verifikasi'
, 'kode verifikasix':'kode_verifikasi'
, 'kode verifikasin':'kode_verifikasi'
, 'kode verifikasihnya':'kode_verifikasi'
, 'kode verifikasinya':'kode_verifikasi'
, 'kode verifikasiny':'kode_verifikasi'
, 'kode verifikasix':'kode_verifikasi'
, 'kode verifikasin':'kode_verifikasi'
, 'kode verifikasihnya':'kode_verifikasi'
, 'kode verifikasinya':'kode_verifikasi'
, 'kode verifikasiny':'kode_verifikasi'
, 'kode verifikasix':'kode_verifikasi'
, 'kode verifikasin':'kode_verifikasi'
, 'kode verifikasihnya':'kode_verifikasi'
, 'kode verifikasinya':'kode_verifikasi'
, 'kode verifikasiny':'kode_verifikasi'
, 'kode verifikasix':'kode_verifikasi'
, 'kode verifikasin':'kode_verifikasi'
, 'kode verifikasihnya':'kode_verifikasi'

, 'kode_verifikasi_email_lama':'lamban_untuk_mendapatkan_kode_verifikasi'
, 'kode_verifikasi_lama':'lamban_untuk_mendapatkan_kode_verifikasi'
, 'kode_verifikasi_lama':'lamban_untuk_mendapatkan_kode_verifikasi'


```
co72 = {r'(\b){ }(\b)'.format(call72):r'\1{\}\2'.format(wind72) for call72, wind72 in
kp72.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co72, regex=True)
```

```
kp73 = { 'bukan':"
```

```
    , 'bukanx':"
```

```
    , 'bukany':"
```

```
    , 'bukannya':"
```

```
    , 'bukanya':"
```

```
    , 'bukanny':"
```

```
    , 'bkn':"
```

```
    , 'bknx':"
```

```
    , 'bkny':"
```

```
    , 'bknyy':"
```

```
    , 'bknya':"
```

```
    , 'bknyya':"
```

```
    , 'bkan':"
```

```
    , 'bkanx':"
```

```
    , 'bkany':"
```

```
    , 'bkanny':"
```

```
    , 'bkanya':"
```

```
    , 'bkannya':"
```

```
    , 'bukn':"
```

```
    , 'buknx':"
```

```
    , 'bukny':"
```

```
    , 'buknyy':"
```

```
    , 'buknya':"
```

```
    , 'buknyya':"
```




```

, 'aplikasi_dulu': 'aplikasi_versi_dulu'
, 'aplikasi_dulu': 'aplikasi_versi_dulu'
, 'password': 'sandi'

, 'silahkan_ulangi': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'
, 'beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'

, 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'
, 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'

, 'terjadi_kesalahan_terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'
, 'terjadi_kesalahan_terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'

, 'terjadi_kesalahan_terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'
, 'terjadi_kesalahan_terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'

}

```

```

co74 = {r'(\b){(\b)'.format(call74):r'\1{ }\2'.format(wind74) for call74, wind74 in
kp74.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co74, regex=True)

```


, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
beberapa_saat'

, 'terjadi_kesalahan silahkan
 terjadi_kesalahan_silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_
 beberapa_saat'

, 'balikin':"
 , 'dibalikin':"

, 'email dan': 'email_dan'
 , 'email dan': 'email_dan'
 , 'email dan': 'email_dan'
 , 'email dan': 'email_dan'
 , 'email dan': 'email_dan'
 , 'email dan': 'email_dan'
 , 'email dan': 'email_dan'
 , 'email dan': 'email_dan'

, 'email dan_sandi': 'email_dan_sandi'
 , 'email dan_sandi': 'email_dan_sandi'
 , 'email dan_sandi': 'email_dan_sandi'
 , 'email dan_sandi': 'email_dan_sandi'
 , 'email dan_sandi': 'email_dan_sandi'
 , 'email dan_sandi': 'email_dan_sandi'
 , 'email dan_sandi': 'email_dan_sandi'

, 'email dan_pin': 'email_dan_pin'
 , 'email dan_pin': 'email_dan_pin'
 , 'email dan_pin': 'email_dan_pin'
 , 'email dan_pin': 'email_dan_pin'
 , 'email dan_pin': 'email_dan_pin'
 , 'email dan_pin': 'email_dan_pin'
 , 'email dan_pin': 'email_dan_pin'

, 'benar2': 'benar'

```

, 'bnr': 'benar'
, 'bnar': 'benar'
, 'benr': 'benar'
, 'bnr2': 'benar'
, 'bnar2': 'benar'
, 'benr2': 'benar'
, 'bener': 'benar'
, 'bner': 'benar'
, 'benr': 'benar'
, 'bener2': 'benar'
, 'bner2': 'benar'
, 'benr2': 'benar'
}

co75 = {r'(\b){}(\b)'.format(call75):r'\1{ }\2'.format(wind75) for call75, wind75 in
kp75.items()}

df411 = df41['Review']=df41['Review'].replace(co75, regex=True)

kp76 = {'email_dan sandi': 'email_dan_sandi'
, 'email_dan sandi': 'email_dan_sandi'
, 'email_dan sandi': 'email_dan_sandi'
, 'email_dan sandi': 'email_dan_sandi'
, 'email_dan sandi': 'email_dan_sandi'

, 'email_dan pin': 'email_dan_pin'
, 'email_dan pin': 'email_dan_pin'
, 'email_dan pin': 'email_dan_pin'
, 'email_dan pin': 'email_dan_pin'
, 'email_dan pin': 'email_dan_pin'

}

```



```
co76 = {r'(\b){ }(\b)'.format(call76):r'\1{ }\2'.format(wind76) for call76, wind76 in
kp76.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co76, regex=True)
```

```
kp77 = { 'email_dan_sandi benar': 'email_dan_sandi_benar'
, 'email_dan_sandi benar': 'email_dan_sandi_benar'
, 'email_dan_sandi benar': 'email_dan_sandi_benar'
, 'email_dan_sandi benar': 'email_dan_sandi_benar'
, 'email_dan_sandi benar': 'email_dan_sandi_benar'
, 'email_dan_sandi benar': 'email_dan_sandi_benar'

, 'email_dan_pin benar': 'email_dan_pin_benar'
, 'email_dan_pin benar': 'email_dan_pin_benar'
, 'email_dan_pin benar': 'email_dan_pin_benar'
, 'email_dan_pin benar': 'email_dan_pin_benar'
, 'email_dan_pin benar': 'email_dan_pin_benar'

}

```

```
co77 = {r'(\b){ }(\b)'.format(call77):r'\1{ }\2'.format(wind77) for call77, wind77 in
kp77.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co77, regex=True)
```

```
kp78 = { 'padahal email_dan_sandi_benar': 'padahal_email_dan_sandi_benar'
, 'padahal email_dan_pin_benar': 'padahal_email_dan_pin_benar'
, 'email_dan_sandi_benar': 'padahal_email_dan_sandi_benar'
, 'email_dan_pin_benar': 'padahal_email_dan_pin_benar'

, 'padahal memasukkan': 'padahal_memasukkan'
}

```


, 'dipesanya':"

, 'pesany':"

, 'dipesany':"

, 'pesanx':"

, 'dipesanx':"

, 'pesa':"

, 'pesanya':"

, 'pesaya':"

, 'pesay':"

, 'pesax':"

, 'pesn':"

, 'pesnnya':"

, 'pesnya':"

, 'pesny':"

, 'pesnx':"

, 'psan':"

, 'psannya':"

, 'psanya':"

, 'psany':"

, 'psanx':"

, 'dipesn':"

, 'dipesnnya':"

, 'dipesnya':"

, 'dipesny':"

, 'dipesnx':"

, 'dipsan':"

, 'dipsannya':"

, 'dipsanya':"

```

, 'dipsany':"
, 'dipsanx':"

, 'dipesa':"
, 'dipesanya':"
, 'dipesaya':"
, 'dipesay':"
, 'dipesax':"

}

co79 = {r'(\b){}(\b)'.format(call79):r'\1{}\2'.format(wind79) for call79,wind79 in
kp79.items()}

df411 = df41['Review']=df41['Review'].replace(co79, regex=True)

kp80 = {'login_email_tidak_terdaftar':'login_email_tidak_terdaftar'
, 'login_email':'login_email'
, 'susah_login_email_tidak_terdaftar':'susah_login_email_tidak_terdaftar'
, 'susah_login_email':'susah_login_email'

, 'saat_login':'saat_login'
, 'saat login':'saat_login'
, 'saat login':'saat_login'
, 'saat login':'saat_login'
, 'saat login':'saat_login'
, 'saat login':'saat_login'
, 'saat login':'saat_login'
, 'saat login':'saat_login'

, 'saat_login_tapi':'saat_login_tapi'
, 'saat_login_tapi':'saat_login_tapi'
, 'saat_login_tapi':'saat_login_tapi'

```

```

, 'saat login_tapi': 'saat_login_tapi'
, 'saat login_tapi': 'saat_login_tapi'
, 'saat login_tapi': 'saat_login_tapi'
, 'saat login_tapi': 'saat_login_tapi'

, 'saat login_error': 'saat_login_error'
, 'saat login_error': 'saat_login_error'
, 'saat login_error': 'saat_login_error'
, 'saat login_error': 'saat_login_error'
, 'saat login_error': 'saat_login_error'
, 'saat login_error': 'saat_login_error'

, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'
, 'disaat login': 'saat_login'

, 'padasaat login': 'saat_login'
, 'padasaat login': 'saat_login'
, 'padasaat login': 'saat_login'
, 'padasaat login': 'saat_login'
, 'padasaat login': 'saat_login'
, 'padasaat login': 'saat_login'
, 'padasaat login': 'saat_login'

}

```

```

co80 = {r'(\b){ }(\b)'.format(call80):r'\1{ }2'.format(wind80) for call80, wind80 in
kp80.items()}

```

df411 = df41['Review']=df41['Review'].replace(co80, regex=True)

kp80 = {'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'tidak_bisa aplikasi':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'aplikasi tidak_bisa':'aplikasi_tidak_bisa'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'email tidak_terdaftar':'email_tidak_terdaftar'
 , 'padahal tidak_bisa':'tidak_bisa'
 , 'padahal tidak_bisa':'tidak_bisa'
 , 'padahal tidak_bisa':'tidak_bisa'
 , 'padahal tidak_bisa':'tidak_bisa'


```

, 'padahal tidak_bisa':'tidak_bisa'
, 'padahal tidak_bisa':'tidak_bisa'
, 'padahal tidak_bisa':'tidak_bisa'

, 'padahal email':'padahal_email'
, 'padahal email':'padahal_email'
, 'padahal email':'padahal_email'
, 'padahal email':'padahal_email'
, 'padahal email':'padahal_email'
, 'padahal email':'padahal_email'
, 'padahal email':'padahal_email'

, 'bentar':'benar'

}

co80 = {r'(\b){}(\b)'.format(call80):r'\1{}'\2'.format(wind80) for call80,wind80 in
kp80.items()}

df411 = df41['Review']=df41['Review'].replace(co80, regex=True)

kp81 = {'padahal_email sandi':'padahal_email_sandi'
, 'padahal_email sandi':'padahal_email_sandi'
, 'padahal_email sandi':'padahal_email_sandi'
, 'padahal_email sandi':'padahal_email_sandi'
, 'padahal_email sandi':'padahal_email_sandi'
, 'padahal_email sandi':'padahal_email_sandi'
, 'padahal_email sandi':'padahal_email_sandi'

, 'suda':"
, 'adh':"
, 'loginya':'login'
, 'loginx':'login'

```



```

, 'padahal      dulu':'padahal_dulu'

, 'padahal dulunya':'padahal_dulu'
, 'padahal dulunya':'padahal_dulu'
, 'padahal dulunya':'padahal_dulu'
, 'padahal dulunya':'padahal_dulu'
, 'padahal dulunya':'padahal_dulu'
, 'padahal dulunya':'padahal_dulu'
, 'padahal dulunya':'padahal_dulu'

, 'padahal dulu2nya':'padahal_dulu'
, 'padahal dulu2nya':'padahal_dulu'
, 'padahal dulu2nya':'padahal_dulu'
, 'padahal dulu2nya':'padahal_dulu'
, 'padahal dulu2nya':'padahal_dulu'
, 'padahal dulu2nya':'padahal_dulu'
, 'padahal dulu2nya':'padahal_dulu'

, 'close':'force_close'
, 'kluar':'force_close'

}

co81 = {r'(\b){ }(\b)'.format(call81):r'\1{ }2'.format(wind81) for call81,wind81 in
kp81.items()}

df411 = df41['Review']=df41['Review'].replace(co81, regex=True)

kp82 = {'aplikasi tidak_bisa_dibuka':'aplikasi_bpjstku tidak_bisa_dibuka'
, 'aplikasi tidak_bisa_dibuka':'aplikasi_bpjstku tidak_bisa_dibuka'
, 'aplikasi tidak_bisa_dibuka':'aplikasi_bpjstku tidak_bisa_dibuka'

```



```

, 'aplikasi lama': 'aplikasi_bpjstku_lama'
, 'aplikasi lama': 'aplikasi_bpjstku_lama'
, 'aplikasi lama': 'aplikasi_bpjstku_lama'
, 'aplikasi lama': 'aplikasi_bpjstku_lama'
, 'aplikasi lama': 'aplikasi_bpjstku_lama'
, 'aplikasi lama': 'aplikasi_bpjstku_lama'
, 'aplikasi lama': 'aplikasi_bpjstku_lama'

, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'
, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'
, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'
, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'
, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'
, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'
, 'aplikasi versi_dulu': 'aplikasi_bpjstku_versi_dulu'

}

co82 = {r'(\b){ }(\b)'.format(call82):r'\1{ }2'.format(wind82) for call82, wind82 in
kp82.items()}

df411 = df41['Review'] = df41['Review'].replace(co82, regex=True)

kp83 = { 'aplikasi bpjs': 'aplikasi_bpjstku'
, 'aplikasi bpjs': 'aplikasi_bpjstku'
, 'aplikasi bpjs': 'aplikasi_bpjstku'
, 'aplikasi bpjs': 'aplikasi_bpjstku'
, 'aplikasi bpjs': 'aplikasi_bpjstku'
, 'aplikasi bpjs': 'aplikasi_bpjstku'
, 'aplikasi bpjs': 'aplikasi_bpjstku'

, 'aplikasi bpjstk': 'aplikasi_bpjstku'
, 'aplikasi bpjstk': 'aplikasi_bpjstku'

```

, 'aplikasi bpjstk': 'aplikasi_bpjstku'
 , 'aplikasi bpjstk': 'aplikasi_bpjstku'
 , 'aplikasi bpjstk': 'aplikasi_bpjstku'
 , 'aplikasi bpjstk': 'aplikasi_bpjstku'
 , 'aplikasi bpjstk': 'aplikasi_bpjstku'

, 'aplikasi bpjstku': 'aplikasi_bpjstku'
 , 'aplikasi bpjstku': 'aplikasi_bpjstku'
 , 'aplikasi bpjstku': 'aplikasi_bpjstku'
 , 'aplikasi bpjstku': 'aplikasi_bpjstku'
 , 'aplikasi bpjstku': 'aplikasi_bpjstku'
 , 'aplikasi bpjstku': 'aplikasi_bpjstku'
 , 'aplikasi bpjstku': 'aplikasi_bpjstku'

, 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjsketenagakerjaan': 'aplikasi_bpjstku'

, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
 , 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'

, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'

```
, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs tenagakerjaan': 'aplikasi_bpjstku'
```

```
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
```

```
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
, 'aplikasi bpjs ktenagakerjaan': 'aplikasi_bpjstku'
```

```
}
```

```
co83 = {r'(\b){ }(\b)'.format(call83):r'\1{ }2'.format(wind83) for call83,wind83 in
kp83.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co83, regex=True)
```

```
kp84 = { 'bpjs': "
, 'bpjstk': "
```



```

, 'bpjstku':"
, 'bpjsketenagakerjaan':"
, 'ktenagekerjaan':"
, 'tenagakerjaan':"
, 'ktenagakerjaan':"
, 'aplikasi':"

, 'aplikasi_error':'aplikasi_bpjstku_error'
, 'aplikasi_versi_dulu':'aplikasi_bpjstku_versi_dulu'
, 'tidak_bisa_masuk_aplikasi':'tidak_bisa_masuk_aplikasi_bpjstku'

}

co84 = {r'(\b){ }(\b)'.format(call84):r'\1 { } \2'.format(wind84) for call84,wind84 in
kp84.items()}

df411 = df41['Review']=df41['Review'].replace(co84, regex=True)

kp85 = { 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'
, 'terdaftar terdaftar':'terdaftar'

, 'email_terdaftar terdaftar':'email_terdaftar'
, 'email_terdaftar terdaftar':'email_terdaftar'
, 'email_terdaftar terdaftar':'email_terdaftar'
, 'email_terdaftar terdaftar':'email_terdaftar'

```


, 'update
 tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_mengalami_force_close'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'update
 tidak_bisa_login_email_tidak_terdaftar':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_karena_email_tidak_terdaftar'

, 'dupdate tidak_bisa_login tidak_ada
 email':'setelah_update_aplikasi_bpjstku_tidak_bisa_login_karena_email_tidak_terdaftar',

, 'dupdate tidak_bisa_login tidak_ada
 email':'setelah_update_aplikasi_bpjstku_tidak_bisa_login_karena_email_tidak_terdaftar',

, 'dupdate tidak_bisa_login tidak_ada
email':'setelah_update_aplikasi_bpjstku_tidak_bisa_login_karena_email_tidak_terdaftar
,

, 'dupdate tidak_bisa_login tidak_ada
email':'setelah_update_aplikasi_bpjstku_tidak_bisa_login_karena_email_tidak_terdaftar
,

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'padahal_sudah_di_update
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'padahal_sudah_di_update
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'diupdate
 tidak_bisa_login_email_tidak_terdaftar': 'setelah_update_aplikasi_bpjstku_login_tidak_
 bisa_karena_email_tidak_terdaftar'

, 'terupdate
 tidak_bisa_login_force_close': 'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
 alami_force_close'

, 'terupdate
 tidak_bisa_login_force_close': 'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
 alami_force_close'


```

    , 'diupdate
tidak_bisa_login_force_close':'setelah_update_aplikasi_login_tidak_bisa_mengalami_force_close'
    , 'diupdate
tidak_bisa_login_force_close':'setelah_update_aplikasi_login_tidak_bisa_mengalami_force_close'

}

co88 = {r'(\b){ }(\b)'.format(call88):r'\1{\2}'.format(wind88) for call88,wind88 in
kp88.items()}

df411 = df41['Review']=df41['Review'].replace(co88, regex=True)

kp88 = {'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'

, 'update_tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'update_tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'update_tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'update_tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'update_tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'

```

, 'update tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'update tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'update tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'update tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'setelah_aplikasi_update
 tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'padahal_sudah_di_update
 tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'padahal_sudah_di_update
 tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
 , 'padahal_sudah_di_update
 tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'

, 'padahal_sudah_di_update
tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
, 'padahal_sudah_di_update
tidak_bisa_dibuka':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'

, 'setelah_aplikasi_update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'setelah_aplikasi_update_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'setelah_aplikasi_update
tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'setelah_aplikasi_update
tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'setelah_aplikasi_update
tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'setelah_aplikasi_update
tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'setelah_aplikasi_update
tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'

, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'diupdate_tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'

, 'diupdate_tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
, 'diupdate
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
, 'diupdate
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'
, 'diupdate
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_dibuka'

, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'padahal_sudah_di_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'setelah_aplikasi_update
tidak_bisa_masuk':'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk'
, 'diupdate
tidak_bisa_cek':'setelah_update_aplikasi_bpjstku_tidak_bisa_cek_saldo_jht'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,diupdate tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

,update tidak_bisa_memasukkan
email:'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaft
ar'

, 'update tidak_bisa_memasukkan
email': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update tidak_bisa_memasukkan
email': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update tidak_bisa_memasukkan
email': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update
tidak_bisa_daftar': 'setelah_update_aplikasi_bpjstku_tidak_bisa_masuk_karena_email_tidak_terdaftar'

, 'update tidak_bisa': 'setelah_update_aplikasi_bpjstku_tidak_bisa'


```

, 'dupdate tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'dupdate tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'dupdate tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'dupdate tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'dupdate tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'
, 'dupdate tidak_bisa':'setelah_update_aplikasi_bpjstku_tidak_bisa'

}

co90 = {r'(\b){(\b)}.format(call90):r'\1{\2}'.format(wind90) for call90,wind90 in
kp90.items()}

df411 = df41['Review']=df41['Review'].replace(co90, regex=True)

kp91 = {'terupdate tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'
, 'terupdate tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'
, 'terupdate tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'
, 'terupdate tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'
, 'terupdate
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'
, 'terupdate
tidak_bisa_login':'setelah_update_aplikasi_bpjstku_login_tidak_bisa'

, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'
, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

```

, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'setelah_aplikasi_update
tidak_bisa_login_force_close':'setelah_update_aplikasi_bpjstku_login_tidak_bisa_meng
alami_force_close'

, 'loading_nya_lama':'proses_loading_lama'
, 'loading_nya_lama':'proses_loading_lama'
, 'loading_nya_lama':'proses_loading_lama'

, 'loading_lama':'proses_loading_lama'

, 'loading_lama':'proses_loading_lama'
, 'loading_lama':'proses_loading_lama'
, 'loading_lama':'proses_loading_lama'
, 'loading_lama':'proses_loading_lama'
, 'loading_lama':'proses_loading_lama'

, 'loading_lamban':'proses_loading_lama'
, 'loading_lambat':'proses_loading_lama'

, 'cuma':"

```

, 'loading': 'proses_loading_lama'
, 'loading_nya': 'proses_loading_lama'

, 'cek_saldo_jht proses_loading_lama': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'cek_saldo_jht proses_loading_lama': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'cek_saldo_jht proses_loading_lama': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'cek_saldo_jht
proses_loading_lama': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'cek_saldo_jht
proses_loading_lama': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'cek_saldo_jht
proses_loading_lama': 'proses_loading_lama_untuk_cek_saldo_jht'

, 'proses_loading_lama
cek_saldo_jht': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'proses_loading_lama
cek_saldo_jht': 'proses_loading_lama_untuk_cek_saldo_jht'
, 'proses_loading_lama
cek_saldo_jht': 'proses_loading_lama_untuk_cek_saldo_jht'

}

```

```

co91 = {r'(\b){}(\b)'.format(call91):r'\1{}\2'.format(wind91) for call91, wind91 in
kp91.items()}

```

```

df411 = df41['Review'] = df41['Review'].replace(co91, regex=True)

```

```

kp92 = {'klaim proses_loading_lama': 'e_klaim_proses_loading_lama'
, 'klaim proses_loading_lama': 'e_klaim_proses_loading_lama'
, 'klaim proses_loading_lama': 'e_klaim_proses_loading_lama'
, 'klaim proses_loading_lama': 'e_klaim_proses_loading_lama'
, 'klaim proses_loading_lama': 'e_klaim_proses_loading_lama'

```

```

, 'klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'klaim proses_loading_lama':'e_klaim_proses_loading_lama'

, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'
, 'e_klaim proses_loading_lama':'e_klaim_proses_loading_lama'

, 'mengklaim':'e_klaim'
, 'meng_klaim':'e_klaim'

}

co92 = {r'(\b){ }(\b)'.format(call92):r'\1{ }\2'.format(wind92) for call92,wind92 in
kp92.items()}

df411 = df41['Review']=df41['Review'].replace(co92, regex=True)

kp93 = {'e_klaim_proses_loading_lama':'proses_loading_lama_untuk_e_klaim'
, 'tidak_bisa_login
padahal_email_dan_sandi_benar':'tidak_bisa_login_padahal_email_dan_sandi_benar'
,
'padahal_email_dan_sandi_benar':'tidak_bisa_login_padahal_email_dan_sandi_benar'

, 'cek_saldo_jht klaim':'proses_loading_lama_untuk_e_klaim'
, 'cek_saldo_jht klaim':'proses_loading_lama_untuk_e_klaim'
, 'cek_saldo_jht klaim':'proses_loading_lama_untuk_e_klaim'
, 'cek_saldo_jht klaim':'proses_loading_lama_untuk_e_klaim'
, 'cek_saldo_jht klaim':'proses_loading_lama_untuk_e_klaim'

```

, 'cek_saldo_jht e_klaim':'proses_loading_lama_untuk_e_klaim'
 , 'cek_saldo_jht e_klaim':'proses_loading_lama_untuk_e_klaim'
 , 'cek_saldo_jht e_klaim':'proses_loading_lama_untuk_e_klaim'
 , 'cek_saldo_jht e_klaim':'proses_loading_lama_untuk_e_klaim'
 , 'cek_saldo_jht e_klaim':'proses_loading_lama_untuk_e_klaim'
 , 'cek_saldo_jht e_klaim':'proses_loading_lama_untuk_e_klaim'

, 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'

, 'susah_login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'susah_login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'susah_login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'susah_login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'
 , 'susah_login
 terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saas_login'

```

, 'susah_login
terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saat_login'
, 'susah_login
terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saat_login'

, 'terjadi_kesalahan_server':'muncul_pesan_terjadi_kesalahan_server_saat_login'

, 'susah_login':'muncul_pesan_terjadi_kesalahan_server_saat_login'

, 'elektronik':"

, 'niki':"

, 'elekteonik':"
, 'elktronik':"
, 'elektronik':"
, 'elektonik':"

, 'panik':"

, 'menikmati':"

}

```

```

co93 = {r'(\b){ }(\b)'.format(call93):r'\1{ }\2'.format(wind93) for call93,wind93 in
kp93.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co93, regex=True)

```

```

kp94 = { 'klinik':"
, 'inikan':"
, 'nikah':"

```

```

, 'kelinik':"
, 'komunikasi':"
, 'nikmati':"
, 'nikmatin':"
, 'kenikmtan':"
, 'piknik':"
, 'unik':"
, 'nike':"
, 'penikmat':"
, 'diklinik':"
, 'niklah':'nik'
, 'elektronik':"
, 'nomornik':'nomor nik'

, 'nomor_bpjs':'nomor_bpjstku'

, 'nomor_kartu_bpjs':'nomor_bpjstku'

}

```

```

co94 = {r'(\b){ }(\b)'.format(call94):r'\1 { } \2'.format(wind94) for call94,wind94 in
kp94.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co94, regex=True)

```

```

#KODE VERIFIKASI#

```

```

#KODE

```

```

VERIFIKASI# #KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#

```

```

#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#
#KODE VERIFIKASI#

```

```

kp95 = {'nomor_bpjstku':'nomor_kartu_bpjstku'
,
'kode_verifikasi':'kode_verifikasi_tidak_terkirim_atau_terlalu_lama_ke_pengguna_apli
kasi_bpjstku_sehingga_kode_kadaluwarsa'
}

```

```

co95 = {r'(\b){ }(\b)'.format(call95):r'\1{ }\2'.format(wind95) for call95,wind95 in
kp95.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co95, regex=True)

```

```

kp96 = {'tidak_bisa_buat_akun':'tidak_bisa_buat_akun'
, 'dapat_balasan':"
, 'dapat_balasan':"
, 'dapat_balasan':"
, 'dapat_balasan':"

, 'dapat':"
, 'balasan':"

, 'buat_akun_tidak_bisa':'tidak_bisa_buat_akun'
, 'buat_akun_tidak_bisa':'tidak_bisa_buat_akun'

```


, 'buat_akun tidak_bisa': 'tidak_bisa_buat_akun'
 , 'buat_akun tidak_bisa': 'tidak_bisa_buat_akun'
 , 'buat_akun tidak_bisa': 'tidak_bisa_buat_akun'
 , 'buat_akun tidak_bisa': 'tidak_bisa_buat_akun'
 , 'buat_akun tidak_bisa': 'tidak_bisa_buat_akun'

, 'tidak_bisa buat_akun': 'tidak_bisa_buat_akun'
 , 'tidak_bisa buat_akun': 'tidak_bisa_buat_akun'
 , 'tidak_bisa buat_akun': 'tidak_bisa_buat_akun'
 , 'tidak_bisa buat_akun': 'tidak_bisa_buat_akun'

, 'buat_akun gagal': 'tidak_bisa_buat_akun'
 , 'buat_akun gagal': 'tidak_bisa_buat_akun'
 , 'buat_akun gagal': 'tidak_bisa_buat_akun'
 , 'buat_akun gagal': 'tidak_bisa_buat_akun'
 , 'buat_akun gagal': 'tidak_bisa_buat_akun'
 , 'buat_akun gagal': 'tidak_bisa_buat_akun'
 , 'buat_akun gagal': 'tidak_bisa_buat_akun'

, 'lebih_baik':"
 , 'susah_buat_akun': 'tidak_bisa_buat_akun'
 , 'gagal_buat_akun': 'tidak_bisa_buat_akun'
 , 'buat_akun sulittt': 'tidak_bisa_buat_akun'

, 'buat_akun susah': 'tidak_bisa_buat_akun'
 , 'buat_akun susah': 'tidak_bisa_buat_akun'
 , 'buat_akun susah': 'tidak_bisa_buat_akun'
 , 'buat_akun susah': 'tidak_bisa_buat_akun'

, 'nomor nomor':"
 , 'nomor nomor':"
 , 'nomor nomor':"
 , 'nomor nomor':"

, 'buat_akun force_close': 'saat_buat_akun_aplikasi_force_close'
, 'buat_akun force_close': 'saat_buat_akun_aplikasi_force_close'

, 'hang': ''
, 'bolak': 'kembali_ke_halaman_awal'
, 'balik': 'kembali_ke_halaman_awal'

, 'akun force_close': 'saat_buat_akun_aplikasi_force_close'
, 'akun force_close': 'saat_buat_akun_aplikasi_force_close'
, 'akun force_close': 'saat_buat_akun_aplikasi_force_close'
, 'akun force_close': 'saat_buat_akun_aplikasi_force_close'
, 'akun force_close': 'saat_buat_akun_aplikasi_force_close'
, 'akun force_close': 'saat_buat_akun_aplikasi_force_close'

, 'akun error': 'akun_tidak_terdaftar'
, 'akun error': 'akun_tidak_terdaftar'
, 'akun error': 'akun_tidak_terdaftar'
, 'akun error': 'akun_tidak_terdaftar'
, 'akun error': 'akun_tidak_terdaftar'
, 'akun error': 'akun_tidak_terdaftar'
, 'akun error': 'akun_tidak_terdaftar'

, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'
, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'
, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'
, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'
, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'
, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'
, 'akun tidak_bisa_masuk': 'akun_tidak_terdaftar'

, 'akun tidak_ada': 'akun_tidak_terdaftar'
, 'akun tidak_ada': 'akun_tidak_terdaftar'
, 'akun tidak_ada': 'akun_tidak_terdaftar'

, 'tidak_bisa daftar akun': 'tidak_bisa_daftar_akun'

, 'tidak_bisa_daftar akun': 'tidak_bisa_daftar_akun'

, 'tidak_bisa_daftar akun': 'tidak_bisa_daftar_akun'

, 'tidak_bisa_daftar akun': 'tidak_bisa_daftar_akun'

, 'tidak_bisa_daftar akun': 'tidak_bisa_daftar_akun'

, 'tidak_bisa_daftar akun': 'tidak_bisa_daftar_akun'

, 'tidak_bisa_daftar akun': 'tidak_bisa_daftar_akun'

}

co98 = {r'(\b){ }(\b)'.format(call98):r'\1{ }2'.format(wind98) for call98, wind98 in kp98.items() }

df411 = df41['Review'] = df41['Review'].replace(co98, regex=True)

kp99 = { 'padahal akun terdaftar': 'padahal_akun_terdaftar'

, 'padahal akun terdaftar': 'padahal_akun_terdaftar'

, 'padahal akun terdaftar': 'padahal_akun_terdaftar'

, 'padahal akun terdaftar': 'padahal_akun_terdaftar'

, 'padahal akun terdaftar': 'padahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

, 'tidak_bisa_login akun': 'tidak_bisa_login_padaahal_akun_terdaftar'

```

    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login
akun_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_akun_tidak_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
    , 'tidak_bisa_login akun_yang_dulu':'tidak_bisa_login_padaahal_akun_terdaftar'
}

```

```

co99 = {r'(\b){ }(\b)'.format(call99):r'\1{ }2'.format(wind99) for call99,wind99 in
kp99.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co99, regex=True)

```



```

    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_tidak_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
    , 'akun
email_terdaftar':'tidak_bisa_login_muncul_pesan_email_tidak_terdaftar'
}

```

```

co100 = {r'(\b){}\b'.format(call100):r'1{}2'.format(wind100) for call100,wind100 in
kp100.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co100, regex=True)

```

```

kp101 = {'saat':"

```

, 'isi':"
, 'ngisi':"
, 'dapet':"
, 'salahnya':"
, 'dimana':"
, 'tulis':"
, 'knpa':"
, 'mqu':"
, 'temukan':"
, '2x':"
, 'tng':"
, 'bpk':"
, 'bu':"
, 'dulu':"
, 'pernah':"
, 'trss':"
, 'eror2': 'error'
, 'bin':"
, 'yth':"
, 'bpak':"
, 'emil': 'email'
, 'prnah':"
, 'pertama':"
, 'penerima':"
, 'disuru':"
, 'kolom':"
, 'perbarui':"
, 'perkembangan':"
, 'kebangetan':"
, 'cs':"
, 'csnya':"
, 'sibuk':"
, 'melulu':"



, 'tahu':"

, 'volume':"

, 'kedengaran':"

, 'suara':"

, 'dering':"

, 'deringnya':"

, 'mental':'force_close'

, 'gt':"

, 'jawabannya':"

, 'dan':"

, 'pc':"

, 'knp2':"

, 'langsung':"

, 'terulang':'kembali_ke_halaman_awal'

, 'ulang':'kembali_ke_halaman_awal'

, 'kira2':"

, 'petunjuk':"

, 'ehh':"

, 'tapo':"

, 'hasilnya':"

, 'nihil':"

, 'antara':"

, 'habis':"

, 'emaosi':"

, 'komplain':"

, 'jawaban':"

, 'suruh':"

, 'mala':"

, 'cb':"

, 'perlu':"

, 'waspada':"

, 'waspada':"

, 'pakai':"

```

, 'solusi':"
, 'jawab':"
, 'lanjut':"
, 'sellu':"
, 'ajh':"
, 'bget':"
, 'resign':"
, 'cuman':"
, 'coy':"
, 'bertanya':"
, 'yahh':"
, 'mengakses':"
}

```

```

co101 = {r'(\b){(\b)'.format(call101):r'\1{\2}'.format(wind101) for call101,wind101 in
kp101.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co101, regex=True)

```

```

kp101 = {'emosi':"
, 'berkualitas':"
, 'hal':"
, 'buk':"
, 'blom':"
, 'sj':"
, 'note':"
, 'diulasan':"
, 'dibuang':"
, 'emailny':'email'
, 'gakbisa':'tidak_bisa'

```

, 'besok':"

, 'masuk':"

, 'memasukkan':"

, 'data':"

, 'gila':"

, 'stiap':"

, 'ketik':"

, 'verivikasi': 'verifikasi'

, 'ngk':"

, 'masuk2':"

, 'berulang2': 'berulang'

, 'mencoba':"

, 'salam':"

, 'udahvberulang': 'berulang'

, 'publikasikan':"

, 'ngeblest':"

, 'membantu':"

, 'batas':"

, 'minimum':"

, 'laptop':"

, 'antre':"

, 'ujung2nya':"

, 'udah':"

, 'sda':"

, 'padah':"

, 'dikompir':"

, 'permintaan':"

, 'harap':"

, 'pelayanan':"

, 'masyarakat':"

, 'berhari':"

, 'gag':"

, 'kasihan':"

```

, 'luncurkn':"
, 'season':"
, 'pencet':"
, 'klik':"
, 'tak':"
, 'kasi':"
, 'nol':"
, 'bintang_nol':"
, 'hampir':"
, 'cabang':"
, 'enggak':"
, 'selanjutnya':"
, 'dicoba':"
, 'veryfikasi':'verifikasi'
, 'nungguin':"
, 'ampe':"
, 'dikirim2':"
, 'dftar':'daftar'
, 'berusaha':"
, '175':"
, 'puluhan':"
, 'tingkatkan':"
, 'offline':"
, 'online':"
}

```

```

co101 = {r'(\b){(\b)'.format(call101):r'\1{\}\2'.format(wind101) for call101,wind101 in
kp101.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co101, regex=True)

```

kp102 = {'berhari2':"

 , 'gtu2':"

 , 'gua':"

 , 'ngestuck':"

 , 'sebelum':"

 , 'silahkan':"

 , 'selamat':"

 , 'siang':"

 , 'malam':"

 , 'terhormat':"

 , 'kejelasannya':"

 , 'klu':"

 , 'kesekiankali':"

 , 'fix':"

 , 'penjelasannya':"

 , 'sinyal aman':'sinyal_lancar'

 , 'sinyal aman':'sinyal_lancar'

 , 'sinyal aman':'sinyal_lancar'

 , 'sinyal aman':'sinyal_lancar'

 , 'sinyal aman':'sinyal_lancar'

 , 'sinyal aman':'sinyal_lancar'

 , 'sinyal aman':'sinyal_lancar'

 , 'kemudian':"

 , 'tampilan':"

 , 'berulangkali':'berulang'

 , 'kucing':"

 , 'passwordny':'sandi'

 , 'sja':"

 , 'enakan':"

 , 'selebaran':"

 , 'dbuka':'dibuka'

 , 'ngcek':'cek'

 , 'sgra':"

, 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'

, 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'
 , 'nomor verifikasinya':'kode_verifikasi'

, 'saat memasukkan nomor verifikasi':'saat memasukkan kode_verifikasi'
 , 'saat memasukkan nomor verifikasi':'saat memasukkan kode_verifikasi'
 , 'saat memasukkan nomor verifikasi':'saat memasukkan kode_verifikasi'
 , 'saat memasukkan nomor verifikasi':'saat memasukkan kode_verifikasi'
 , 'saat memasukkan nomor verifikasi':'saat memasukkan kode_verifikasi'

,

'kode_verifikasi_tidak_terkirim_atau_terlalu_lama_ke_pengguna_aplikasi_bpjstku_sehingga_kode_kadaluwarsa':'pengguna_tidak_dapat_kode_verifikasi'

, 'huruf kapital':'selalu_huruf_kapital'
 , 'huruf kapital':'selalu_huruf_kapital'

, 'huruf kapitalnya':'selalu_huruf_kapital'
 , 'huruf kapitalnya':'selalu_huruf_kapital'
 , 'huruf kapitalnya':'selalu_huruf_kapital'
 , 'huruf kapitalnya':'selalu_huruf_kapital'
 , 'huruf kapitalnya':'selalu_huruf_kapital'

, 'nomor_k_p_j_tidak_terdaftar':'nomor_k_p_j_tidak_terdaftar'


```

, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'
, 'nama_sesuai nomor_k_t_p': 'nama_sesuai_k_t_p'

, 'aplikasi_bpjstku_tidak_bisa_dibuka': 'aplikasi_bpjstku_tidak_bisa_dibuka'

, 'aplikasi_tidak_bisa': 'aplikasi_bpjstku_tidak_bisa_dibuka'

, 'aplikasi_force_close': 'aplikasi_bpjstku_force_close'

, 'aplikasi_tidak_bisa_dibuka': 'aplikasi_bpjstku_tidak_bisa_dibuka'

, 'aplikasi_versi_dulu': 'aplikasi_bpjstku_versi_dulu'

, 'lebih_baik_aplikasi_versi_dulu': 'lebih_baik_aplikasi_bpjstku_versi_dulu'

, 'aplikasi_berhenti': 'aplikasi_bpjstku_force_close'

, 'aplikasi_tidak_berfungsi': 'aplikasi_bpjstku_tidak_berfungsi'
}

```

```

co104 = {r'(\b){}\(b)'.format(call104):r'\1{ }\2'.format(wind104) for call104,wind104 in
kp104.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co104, regex=True)

```

```

kp105 = {'klaim': 'e_klaim'}

```

```

, 'eklaim': 'e_klaim'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'
, 'cek_saldo_jht proses_loading_lama': 'cek_saldo_jht_proses_loading_lama'

, 'bgus': ''
, 'aplikasi_bpjstku_versi_dulu': ''
, 'pakai_aplikasi_dulu': ''

, 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat': ''
, 'jaringan_internet': 'jaringan_internet'

}

```

```

co105 = {r'(\b){ }(\b)'.format(call105):r'1{ }2'.format(wind105) for call105, wind105 in
kp105.items()}

```

```

df411 = df41['Review'] = df41['Review'].replace(co105, regex=True)

```

```

kp106 = {'padahal_sinyal_lancar': 'padahal_sinyal_lancar'
, 'padahal_sinyal_lancar': 'padahal_sinyal_lancar'

```

```

, 'login_force_close': 'login_force_close'
, 'login_force_close': 'login_force_close'
, 'login_force_close': 'login_force_close'
, 'login_force_close': 'login_force_close'
, 'login_force_close': 'login_force_close'

```

```

, 'login force_close':'login_force_close'
, 'login force_close':'login_force_close'

, 'saat_login force_close':'saat_login_force_close'
, 'saat_login force_close':'saat_login_force_close'
, 'saat_login force_close':'saat_login_force_close'
, 'saat_login force_close':'saat_login_force_close'
, 'saat_login force_close':'saat_login_force_close'
, 'saat_login force_close':'saat_login_force_close'

, 'update force_close':'setelah_update_force_close'
, 'update force_close':'setelah_update_force_close'
, 'update force_close':'setelah_update_force_close'
, 'update force_close':'setelah_update_force_close'
, 'update force_close':'setelah_update_force_close'
, 'update force_close':'setelah_update_force_close'
, 'update force_close':'setelah_update_force_close'

, 'verifikasi_email':'verifikasi_via_email'
, 'verifikasi_email':'verifikasi_via_email'
, 'verifikasi_email':'verifikasi_via_email'
, 'verifikasi_email':'verifikasi_via_email'
, 'verifikasi_email':'verifikasi_via_email'
, 'verifikasi_email':'verifikasi_via_email'
, 'verifikasi_email':'verifikasi_via_email'

}

```

```

co106 = {r'(\b){ }(\b)'.format(call106):r'\1{ }2'.format(wind106) for call106,wind106 in
kp106.items()}

```

```

df411 = df41['Review']=df41['Review'].replace(co106, regex=True)

```

kp106 = {'verifikasi_via_email':'tidak_menerima_kode_verifikasi_via_email'
 , 'email_email_tidak_terdaftar':'email_tidak_terdaftar'
 , 'email_email_tidak_terdaftar':'email_tidak_terdaftar'
 , 'email_email_tidak_terdaftar':'email_tidak_terdaftar'
 , 'email_email_tidak_terdaftar':'email_tidak_terdaftar'
 , 'email_email_tidak_terdaftar':'email_tidak_terdaftar'
 , 'email_email_tidak_terdaftar':'email_tidak_terdaftar'

 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'
 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'
 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'
 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'
 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'
 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'
 , 'email_tidak_terdaftar_email':'email_tidak_terdaftar'

 , 'tidak_bisa_saldo':'tidak_bisa_cek_saldo_jht'
 , 'tidak_bisa_saldo':'tidak_bisa_cek_saldo_jht'
 , 'tidak_bisa_saldo':'tidak_bisa_cek_saldo_jht'
 , 'tidak_bisa_saldo':'tidak_bisa_cek_saldo_jht'
 , 'tidak_bisa_saldo':'tidak_bisa_cek_saldo_jht'
 , 'tidak_bisa_saldo':'tidak_bisa_cek_saldo_jht'

 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'
 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'
 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'
 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'
 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'
 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'
 , 'saldo_tidak_bisa':'tidak_bisa_cek_saldo_jht'

 , 'terjadi_kesalahan_server':'terjadi_kesalahan_server'

```

, 'terjadi_kesalahan_server': 'terjadi_kesalahan_server'
, 'terjadi_kesalahan_server': 'terjadi_kesalahan_server'
, 'terjadi_kesalahan_server': 'terjadi_kesalahan_server'
, 'terjadi_kesalahan_server': 'terjadi_kesalahan_server'

}

co106 = {r'(\b){}(\b)'.format(call106):r'1{}2'.format(wind106) for call106, wind106 in
kp106.items()}

df411 = df41['Review'] = df41['Review'].replace(co106, regex=True)

kp107 = {'tidak_bisa_login': 'login_gagal'
, 'terjadi_kesalahan_server': 'muncul_pesan_terjadi_kesalahan_server'
, 'terjadi_kesalahan': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'
,
'silahkan_ulangi_beberapa_saat': 'terjadi_kesalahan_silahkan_ulangi_beberapa_saat'
,
'muncul_pesan_terjadi_kesalahan_server_saat_login': 'muncul_pesan_terjadi_kesalahan_
saat_login'

, 'server_error': 'server_error'
, 'server_error': 'server_error'
, 'server_error': 'server_error'
, 'server_error': 'server_error'
, 'server_error': 'server_error'
, 'server_error': 'server_error'

, 'daftar
pengguna_tidak_dapat_kode_verifikasi': 'tidak_bisa_daftar_karena_pengguna_tidak_dap
at_kode_verifikasi'

```



```
}
```

```
co108 = {r'(\b){ }(\b)'.format(call108):r'\1{ }\2'.format(wind108) for call108,wind108 in
kp108.items() }
```

```
df411 = df41['Review']=df41['Review'].replace(co108, regex=True)
```

```
with pd.option_context('display.max_rows', None,'display.max_colwidth', None):
```

```
    display(len(containslidf41('tidak_menerima_kode_verifikasi_via_email')))
```

```
    display(containslidf41('tidak_menerima_kode_verifikasi_via_email'))
```

```
df212=df411
```

```
df212.to_csv('overhauldict.csv', index=False)
```

```
df11 =pd.read_csv('overhauldict.csv', sep=';', header=0)
```

